

PODSTAWY BAZ DANYCH - OPRACOWANIE PYTAŃ Z PIERWSZEGO TERMINU EGZAMINU

//wszystkie uwagi edycyjne zaznaczajcie w ten sposób, żeby od razu rzucało się w oczy co trzeba poprawić

//polecam slajdy: <http://codex.cs.yale.edu/avi/db-book/db5/slide-dir/index.html>

Spis pytań:

1. [21] Postać normalna Boyce'a Codd (+ Co zrobić jeżeli relacja nie spełnia postaci normalnej Boyce'a Codd)
2. [20] Domknięcie atrybutów? 3 zastosowania? Jak go użyć w tych zastosowaniach?
3. [12] Podać 3 aksjomaty Armstronga
4. [8] 3NF
5. [7] Struktura slotowa - przechowywanie rekordów o zmiennej długości
6. [7] Wstawianie do widoków (czego nie można używać?, "with check option")
7. [7] Definicja, rodzaje i zastosowanie widoków
8. [6] Postaci normalne, w tym 4NF
9. [5] Szeregowalność
10. [5] Problemy wielodostępu - każdy problem i podproblem omówić
11. [4] Czym jest rozszerzenie tabeli?
12. [4] Blokady, podejście pesymistyczne i optymistyczne
13. [4] Anomalie i rodzaje anomalii (omówić przykład z wykładu)
14. [4] Co to są nadmiarowe atrybuty i jak się je znajduje?
15. [4] Domknięcie atrybutów - co to jest, pseudokod
16. [4] Mapowanie generalizacji i specjalizacji z modelu związków encji na model relacyjny.
17. [3] Harmonogram sekwencyjny i niesekwencyjny
18. [3] Uprawnienia na widokach
19. [3] Opowiedzieć o kluczach (w kontekście modelu relacyjnego)
20. [3] Kiedy transakcja spełnia warunki blokowania dwufazowego?
21. [3] Funkcjonalne komponenty systemu BD, za co odpowiadają, przetwarzanie zapytań
22. [3] Zależności referencyjne, jak można je modyfikować, kiedy się je definiuje
23. [2] Co to jest agregacja?
24. [2] Co to jest pokrycie kanoniczne (4 warunki)?
25. [2] Dopełnienie zbioru zależności funkcyjnych Z i jak to wyznaczyć?
26. [2] Niedogodności plików do gromadzenia danych
27. [2] REFERENCES permission, ALL permission
28. [2] Kto ma uprawnienia do modyfikacji schematu?
29. [2] Czym jest zbiór związków encji?
30. [2] 2 sposoby dostępu bazy danych pod względem języków programowania
31. [2] Co to jest procedura składowana? Zalety?
32. [2] Algorytm sprawdzania czy dany zbiór atrybutów jest superkluczem - wykorzystaj zbiór domknięć atrybutów
33. [1] Po co tworzymy bazy danych?

34. [1] [Warstwy logiczne i fizyczne bazy danych](#)
35. [1] [Różnica między SQL dynamicznym i osadzonym](#)
36. [1] [Jak się wyznacza klucz główny w zbiorach związków?](#)
37. [1] [Czym są warunki integralnościowe?](#)
38. [1] [Integralność referencyjna](#)
39. [1] [Czy w praktyce klucz obcy może być NULLEM?](#)
40. [1] [Buforowanie bloków danych](#)
41. [1] [Usuwanie kaskadowe w zależnościach referencyjnych](#)
42. [1] [Transformacja atrybutów wielowartościowych, złożonych i pochodnych na model relacyjny](#)
43. [1] [Czym jest klucz kandydujący w kontekście zależności funkcyjnych?](#)
44. [1] [Czym jest klucz kandydujący w ER?](#)
45. [1] [Różnice między własnymi typami, a zawężeniem dziedziny](#)
46. [1] [Generalizacja i specjalizacja w ER](#)
47. [1] [Po co są role i jak je definiujemy?](#)
48. [1] [Konstrukcja obsługi wyjątku](#)
49. [1] [Prawo pierwszeństwa w harmonogramach](#)
50. [1] [Metody optymistyczne/pesymistyczne tworzenia harmonogramu](#)
51. [1] [Zarządzanie buforem](#)
52. [1] [Sandboxing](#)
53. [1] [Zawężenie dziedziny, checki](#)
54. [1] [Formy autoryzacji do modyfikowania schematu bazy](#)
55. [1] [Legalność relacji](#)
56. [1] [Rodzaje języków](#)
57. [1] [Case - co robi?](#)
58. [1] [Algorytm wyliczania dopełnienia \(ogólnie, opisowo\)](#)
59. [1] [Diagram stanów transakcji](#)
60. [1] [Kiedy dekompozycja zachowuje zależność?](#)
61. [1] [Dekompozycja bezstratna + warunki](#)
62. [1] [Czasowe zależności klucza obcego od głównego](#)
63. [1] [Harmonogramy równoważne](#)
64. [1] [Kiedy nie używać triggerów?](#)
65. [1] [Funkcje tablicowe](#)
66. [1] [Co to jest model danych, podaj jego elementy i przykłady modeli. Co to jest model konceptualny? Czym różni się od implementacyjnego?](#)
67. [1] [Funkcje składowania bazy danych](#)
68. [1] [Definicja harmonogramu odtwarzalnego](#)
69. [1] [Jak szeregować harmonogram](#)
70. [1] [Ograniczenie określające, które encje mogą być członkami danego zbioru encji w generalizacji ER](#)
71. [1] [Jak alternatywnie wyliczyć domknięcie zbioru zależności \$F^+\$?](#)
72. [1] [Zależności funkcyjne](#)
73. [1] [Słownik danych - co to jest, co przechowuje, jak jest fizycznie zorganizowany](#)
74. [1] [W rekordzie są 3 pola o zmiennym rozmiarze atrybutów i jeden o stałym, jak to jest fizycznie zorganizowane](#)
75. [1] [Cechy związków encji](#)

- [76. \[1\] Jak projektujemy relacyjną bazę danych](#)
- [77. \[1\] Buffer manager](#)
- [78. \[1\] Zamiana agregacji w ER na postać relacyjną \(na model relacyjny?\)](#)
- [79. \[1\] Jak mają być ułożone transakcje w harmonogramie?](#)
- [80. \[1\] Rodzaje blokad](#)
- [81. \[1\] Co to jest rozszerzenie/redukcja blokady?](#)
- [82. \[1\] Co to instancja? Co to jest schemat?](#)
- [83. \[1\] Czym jest fizyczna izolacja danych?](#)
- [84. \[1\] Migrujące dane w widokach](#)
- [85. \[1\] Czym jest trigger \(definicja\), jak się go definiuje?](#)
- [86. \[1\] Co to jest indeks?](#)
- [87. \[1\] Jak odświeżamy dane w tabeli zmaterializowanej, typy refresh'y](#)
- [88. \[1\] Co to jest procedura skalarna?](#)
- [89. \[1\] O co chodzi z klauzulą with w procedurze?](#)
- [90. \[1\] Co to jest fantom?](#)
- [91. \[1\] Duże pliki danych \(clob/blob\)](#)

1. [21] Postać normalna Boyce'a Codd (+ Co zrobić jeżeli relacja nie spełnia postaci normalnej Boyce'a Codd)

Schemat relacji R jest w BCNF w odniesieniu do zbioru zależności funkcyjnych F jeżeli dla wszystkich zależności funkcyjnych w F+ postaci $\alpha \rightarrow \beta$ gdzie $\alpha \subseteq R$ i $\beta \subseteq R$, zachodzi przynajmniej jeden z warunków:

- $\alpha \rightarrow \beta$ jest trywialną zależnością (tzn. $\beta \subseteq \alpha$)
- α jest superkluczem w R

Jeżeli schemat nie w BCNF to dekomponujemy do mniejszych schematów, które będą w BCNF. Przykład ze slajdów:

inst_dept (ID, name, salary, dept_name, building, budget) (nie jest w BCNF bo zależność funkcyjna dept_name \rightarrow budget

zachodzi na inst_dept, ale dept_name nie jest superkluczem)

dekomponujemy do:

- instructor z zależnością funkcyjną: ID \rightarrow name, dept_name, salary
- department z zależnością funkcyjną: dept_name \rightarrow building, budget

One są już w BCNF

We now state a general rule for decomposing that are not in BCNF . Let R be a schema that is not in BCNF . Then there is at least one nontrivial functional dependency $\alpha \rightarrow \beta$ such that α is not a superkey for R. We replace R in our design with two schemas:

- $(\alpha \cup \beta)$
- $(R - (\beta - \alpha))$

2. [20] Domknięcie atrybutów? 3 zastosowania? Jak go użyć w tych zastosowaniach?

Mając dany zbiór atrybutów A, definiuje się domknięcie zbioru A nad F (oznaczone jako A+) jako zbiór atrybutów, które są funkcyjnie określone przez A nad F.

Zastosowania:

- testowanie superklucza
- testowanie zależności funkcyjnych
- alternatywne wyliczenie domknięcia F+

Testowanie superklucza: Dla domniemanego superklucza wyliczamy domknięcie. Jeżeli jest nim cała tabela, to faktycznie mamy superklucz.

Testowanie zależności funkcyjnych:

Sprawdzamy czy $\alpha \rightarrow \beta$ zachodzi przez sprawdzenie czy $\beta \subseteq \alpha^+$.

Alternatywne wyliczenie domknięcia F+:

Dla każdego $A \subseteq R$ obliczamy domknięcie A^+ , a następnie dla każdego $S \subseteq A^+$, dodajemy do F+ zależność $A \rightarrow S$

3. [12] Podać 3 aksjomaty Armstronga

reguły zwrotności (if $\beta \subseteq \alpha$, then $\alpha \rightarrow \beta$)

reguła rozszerzalności (if $\alpha \rightarrow \beta$, then $\gamma\alpha \rightarrow \gamma\beta$)

reguła przechodniości (if $\alpha \rightarrow \beta$ and $\beta \rightarrow \gamma$, then $\alpha \rightarrow \gamma$)

Pozostałe zasady, które da się z nich wyprowadzić:

- pseudoprzechodniość (if $\alpha \rightarrow \beta$ and $\gamma\beta \rightarrow \delta$, then $\alpha\gamma \rightarrow \delta$)
- unia (if $\alpha \rightarrow \beta$ and $\alpha \rightarrow \gamma$, then $\alpha \rightarrow \beta\gamma$)
- dekompozycja (if $\alpha \rightarrow \beta\gamma$, then $\alpha \rightarrow \beta$ and $\alpha \rightarrow \gamma$)

4. [8] 3NF

1. $\alpha \rightarrow \beta$ jest zależnością trywialną
2. (lub) α jest superkluczem relacji
3. (lub) Każdy atrybut w $\beta - \alpha$ należy do jakiegoś klucza kandydującego

5. [7] Struktura slotowa - przechowywanie rekordów o zmiennej długości

- nagłówek Slotted Page zawiera:
 - * liczba rekordów
 - * koniec wolnej przestrzeni w bloku
 - * lokalizację i rozmiar każdego rekordu
- po nim jest wolna przestrzeń
- potem są rekordy (przy usunięciu jednego przesuwane do końca)
- brak konieczności przechowywania wskaźników bezpośrednio do rekordów

6. [7] Wstawianie do widoków (czego nie można używać?, "with check option")

Generalnie nie należy wstawiać do widoków. SQL udostępnia formalną definicję sytuacji, w których modyfikowanie widoku jest dozwolone. Widok jest modyfikowalny jeśli jego definicja spełnia warunki:

- w klauzuli SELECT nie wolno aliasować atrybutów (~~select ID as 'numer'~~)
- w klauzuli SELECT muszą pojawić się wszystkie atrybuty, które nie są NULLable (czyli mogące być nullami) lub nie posiadają wartości domyślnych
- w klauzuli FROM może występować tylko 1 relacja (i tylko 1 raz)
- brak klauzul GROUP BY i HAVING

7. [7] Definicja, rodzaje i zastosowanie widoków

Widok - dowolna relacja, która nie jest częścią modelu konceptualnego, ale jest widoczna dla użytkownika jako "wirtualna relacja"

Widok dostarcza mechanizmu do ukrywania pewnych danych przed określonym użytkownikiem.

Rodzaje:

- zmaterializowany
- niezmaterializowany

Materializowanie widoku - stworzenie fizycznej relacji zawierającej wszystkie krotki będące wynikiem zapytania definiującego widok; jeżeli relacja użyta w zapytaniu zostanie

zmodyfikowana, widok zmaterializowany staje się nieaktualny

```
CREATE MATERIALIZED VIEW mv_Customer  
AS SELECT * FROM Customer
```

Zalety i wady widoków zmaterializowanych:

Zalety	Wady
- szybszy dostęp do złożonych złączeń	- koszty zarządzania widokiem (np. aktualizacji)
- przydatne przy podsumowywaniu danych	- koszt przechowywania danych na dysku

8. [6] Postaci normalne, w tym 4NF

BCNF (3.5NF) – relacja, w której każdy wyznacznik zależności jest kluczem kandydującym

BCNF ze slajdów:

Schemat relacji R jest w BCNF w odniesieniu do zbioru zależności funkcyjnych F jeśli dla wszystkich zależności funkcyjnych w F+ (domknięciu F) postaci $a \rightarrow b$, gdzie a i b zawierają się w R zachodzi jeden z warunków:

- 1) $a \rightarrow b$ jest zależnością trywialną (b zawiera się w a)
- 2) a jest superkluczem w R

(w uproszczeniu R jest w BCNF w odniesieniu do zbioru zależności funkcyjnych F, jeśli lewa strona każdej nietrywialnej zależności funkcyjnej jest superkluczem)

4NF - relacja w BCNF, która nie zawiera żadnych nietrywialnych zależności wielowartościowych

4NF z db-book: Relacja jest w BCNF w odniesieniu do zbioru zależności funkcyjnych i wielowartościowych D jeśli dla każdej wielowartościowej zależności w domknięciu D (D+) postaci $a \twoheadrightarrow b$, gdzie a i b zawierają się w R spełniony jest jeden z warunków:

- 1) $a \twoheadrightarrow b$ jest trywialne (b zawiera się w a lub a suma b jest równe R)
- 2) a jest superkluczem w R

(w uproszczeniu R jest w 4NF, jeśli każda nietrywialna ZW jest ZF z superkluczem po lewej stronie [jest to uogólnienie BCNF])

9. [5] Szeregowalność

Pojęcie pozwalające rozpoznać wykonania transakcji, które na pewno nie zaburzają spójności bazy. Zadanie szeregowalności polega na rozpoznaniu harmonogramów niesekwencyjnych, które pozwalają wykonywać transakcje współbieżnie, nie dopuszczając do niepożądanych oddziaływań pomiędzy nimi i doprowadzając bazę do stanu, który mógłby równie dobrze powstać w efekcie wykonania sekwencyjnego.

10. [5] Problemy wielodostępu - każdy problem i podproblem omówić

(slajdy 14-18 wykład Transakcje)

-Utrata zmian

Wynik poprawnie zakończonej operacji modyfikacji może zostać przesłonięty przez modyfikację innego użytkownika.

-Niezatwierdzone zależności ("brudne dane")

Gdy jedna transakcja może widzieć pośrednie wyniki tworzone przez inną transakcję, zanim ta zostanie wypełniona. (innymi słowami: zmodyfikowana przez t1 wartość zostaje przeczytana przez t2. Kiedy t1 wykonuje rollback, zmodyfikowana (błędna) wartość wciąż jest używana w t2)

-Analiza niespójności

Pojawia się, gdy transakcja odczytuje kilka wartości z bazy danych, a inna transakcja w tym czasie modyfikuje niektóre z tych wartości (ten problem może wystąpić nawet w transakcji, która jedynie odczytuje dane bez zapisywania)

-Mylące odczyty

Transakcja odczytuje ponownie wartość danej, która została w międzyczasie zmodyfikowana przez inną transakcję

-Odczyty fantomowe

Transakcja wyszukuje w relacji zbiór krotek spełniających warunek. Przy powtórным obliczeniu wyniku pojawiają się dodatkowe, fantomowe krotki, wynik działania innej transakcji.

11. [4] Czym jest rozszerzenie tabeli?

(Slajd 79 SQL_basic_intermediate)

Przy złożonych zapytaniach wygodnie jest przechowywać wynik zapytania jako nową tabelę (tabelę chwilową [temporary]). Tabela zawierająca wyniki zapytania:

CREATE TABLE temp AS (SELECT ...) with data;

lub

CREATE TABLE temp LIKE tabela

12. [4] Blokady, podejście pesymistyczne i optymistyczne

Metody pesymistyczne:

-blokady

-znaczniki czasowe

Metody optymistyczne:

Walidacja. Wykonanie transakcji podzielone na trzy fazy:

- transakcja jest wykonywana i zapisuje dane w zmiennych lokalnych

- transakcja wykonuje test walidacji, który sprawdza, czy zmienne lokalne mogą być zapisane bez naruszenia integralności BD

- jeśli walidacja przebiegła pomyślnie zmienne lokalne są zapisywane w BD, w przeciwnym wypadku następuje rollback

Blokada – zmienna skojarzona z każdą daną w bazie określająca dostępność danej ze względu na możliwość wykonania na niej określonej operacji.

13. [4] Anomalie i rodzaje anomalii (omówić przykład z wykładu)

Anomalie - problemy powstające w przypadku, gdy chcemy umieścić zbyt dużo informacji w jednej relacji.

- redundancja (gdy dane powtarzają się - nadmiarowość danych)
- anomalie wprowadzania danych (np. gdy mamy tabele 'klient' i 'towary', które są reprezentowane jako jedna tabela i chcemy wstawić klienta, to musimy uzupełnić jeszcze inne pola)
- anomalie usuwania danych (np. usuwając klienta, musimy usunąć też powiązane w danej krotce towary)
- anomalie aktualizacji danych (np. aktualizując towary, musimy zrobić to dla każdego klienta)

14. [4] Co to są nadmiarowe atrybuty i jak się je znajduje?

Dla zależności $\alpha \rightarrow \beta$ w F

* Atrybut $A \in \alpha$ jest nadmiarowy w α jeżeli

F logicznie implikuje $(F - \{\alpha \rightarrow \beta\}) \cup \{(\alpha - A) \rightarrow \beta\}$

* Atrybut $B \in \beta$ jest nadmiarowy w β jeżeli

$(F - \{\alpha \rightarrow \beta\}) \cup \{\alpha \rightarrow (\beta - B)\}$ logicznie implikuje F

(ten zbiór różni się od F tym, że z zależności funkcyjnej $\alpha \rightarrow \beta$ zostaje wykluczony atrybut nadmiarowy)

Efektywne testowanie czy atrybut jest nadmiarowy:

* Aby sprawdzić czy atrybut $A \in \alpha$ jest nadmiarowy w α

definiujemy $\gamma = \alpha - \{A\}$

sprawdź, czy $\gamma \rightarrow \beta$ może być wyprowadzone z F:

-wylicz γ + (domknięcie γ) nad F

-jeżeli γ + zawiera wszystkie atrybuty w β , to A jest nadmiarowe w α

* Aby sprawdzić czy atrybut $B \in \beta$ jest nadmiarowy w β

rozważ zbiór $F' = (F - \{\alpha \rightarrow \beta\}) \cup \{\alpha \rightarrow (\beta - B)\}$

sprawdź czy $\alpha \rightarrow B$ może być wywnioskowane z F' :

-wylicz α + (domknięcie α) nad F'

-jeżeli α + zawiera B, to B jest nadmiarowe w β

15. [4] Domknięcie atrybutów - co to jest, pseudokod

Atrybut B jest funkcyjnie zależny (określony) od α jeżeli $\alpha \rightarrow B$

Mając dany zbiór atrybutów α , definiuje się domknięcie zbioru α nad F (oznaczane jako α^+)

jako zbiór atrybutów, które są funkcyjnie określone przez α nad F/

Algorytm do obliczania α^+ , domknięcia zbioru α nad F:

wynik := α

powtarzaj

dla każdej zależności funkcyjnej z F:

jeśli atrybuty będące po lewej stronie zależności należą do wyniku

to do wyniku dodaj atrybuty będące po prawej stronie zależności

dopóki wynik się nie zmienia [w iteracji nie został dodany żaden nowy atrybut]

16. [4] Mapowanie generalizacji i specjalizacji z modelu związków encji na model relacyjny. Klucze główne w relacjach niższych i wyższych poziomów. Generalizacja w ER

Generalizacja łączy kilka zbiorów encji, które mają te same cechy w zbiorze encji wyższego poziomu. Specjalizacja to inwersja generalizacji.

Związek "IS A", zwany również superclass-subclass, oznacza przynależność encji niższego poziomu do encji wyższego poziomu (przykład z db-book: *instructor* "is a" *person*).

Reprezentacja w schemacie

Metoda 1:

- Stwórz schemat dla zbioru encji na wyższym poziomie
- Stwórz schemat dla każdego zbioru encji niższego poziomu włączając PK ze zbioru pojęć wyższego poziomu oraz lokalnych atrybutów
- Dla każdego schematu niższego poziomu stwórz FK
- Wada: wyszukanie informacji o encji niższego poziomu wymaga dostępu do dwóch relacji, odpowiadających encjom niższego i wyższego poziomu

Metoda 2:

- Stwórz schemat dla każdego zbioru encji ze wszystkimi lokalnymi i dziedziczonymi atrybutami
- Jeżeli specjalizacja jest rozłączna i zupełna, schemat dla bardziej ogólnego zbioru pojęć nie wymaga przechowywania informacji
- Wada: może wystąpić redundancja, jeśli obiekt należy do więcej niż jednej encji niższego poziomu

17. [3] Harmonogram sekwencyjny i niesekwencyjny

Harmonogram sekwencyjny składa się z ciągu akcji transakcji, które się nie przeplatają, tzn. najpierw wykonywana jest cała jedna transakcja, potem druga itd.

Harmonogram niesekwencyjny zawiera przeplatające się elementy obu transakcji, ale kolejność w obrębie transakcji jest zachowana. -> mniej więcej tak odpowiedziałam i to była odpowiedź na 5.0

18. [3] Uprawnienia na widokach

Twórca widoku nie dostaje wszystkich uprawnień do niego.

Przyznawanie uprawnień do widoku:

grant select on <nazwa widoku> to <grupa użytkowników>

Formy autoryzacji (uprawnienia) na częściach bazy danych :

Read – pozwala na czytanie, ale nie na modyfikacje danych

Insert – pozwala na wstawianie nowych danych, ale nie na modyfikacje istniejących danych.

Update – pozwala na modyfikacje, ale nie na usuwanie danych.

Delete – pozwala na usuwanie danych.

19. [3] Opowiedzieć o kluczach (w kontekście modelu relacyjnego)

Superklucz - atrybut lub zbiór atrybutów, które razem jednoznacznie identyfikują krotkę w relacji (wartości pozostałych atrybutów).

Klucz kandydujący (potencjalny) - superklucz o najmniejszej ilości atrybutów (może być ich kilka).

Klucz główny/podstawowy (primary key, PK) - wybrany potencjalny klucz do identyfikowania krotek w obrębie danej relacji. Żadne 2 krotki w danej relacji nie mogą mieć takich samych wartości w atrybutach klucza w danym momencie

Klucz obcy relacji (foreign key FK)

- atrybut (lub zbiór atrybutów), który wskazuje na klucz podstawowy
- służy do reprezentowania powiązań między danymi (łączenia relacji)
- dziedziną wartości FK jest dziedzina wartości PK na który FK wskazuje

20. [3] Kiedy transakcja spełnia warunki blokowania dwufazowego?

Transakcja przestrzega protokołu blokowania dwufazowego (2PL) jeżeli wszystkie operacje zakładania blokady znajdują się w niej przed pierwszą operacją zdjęcia blokady.

faza wzrostu - zakładanie i ew. rozszerzanie blokad

faza zmniejszania - zdejmowanie i ew. redukowanie blokad

Twierdzenie

“Jeżeli wszystkie transakcje spełniają protokół dwufazowego blokowania, to wszystkie harmonogramy niesekwencyjne są szeregowalne”

21. [3] Funkcjonalne komponenty systemu BD, za co odpowiadają, przetwarzanie zapytań

1) Menadżer składowania - moduł DBMS, który dostarcza interfejs między danymi niskiego poziomu gromadzonymi w BD a programami aplikacji i pytaniami wysyłanymi do systemu
Składa się z:

- menadżera autoryzacji i integracji
- menadżera transakcji
- menadżera plików
- menadżera buforów

2) Procesor przetwarzania zapytań - zawiera komponenty:

- DDL interpreter
- DML kompilator
- silnik oceny zapytań

22. [3] Zależności referencyjne, jak można je modyfikować, kiedy się je definiuje ze slajdów (nie wiem czy o to chodzi?)

SQL basic 69 slajd: nagłówek - Zależność referencyjna

Gwarantuje że wartość, która występuje w jednej relacji dla danego zestawu atrybutów występuje także dla określonego zestawu atrybutów w innej relacji.

Przykład: Jeżeli “Biology” jest nazwą departamentu, który występuje w jednej z krotek relacji instructor, to istnieje krotka w relacji department dla “Biology”.

Niech A będzie zbiorem atrybutów. Niech R i S będą dwoma relacjami zawierającymi atrybuty A i A jest kluczem głównym w S. Mówi się, że A jest kluczem obcym w R jeżeli dla dowolnej wartości A występującej w R te wartości występują również w S.

23. [2] Co to jest agregacja?

Agregacja - abstrakcja, przez którą zależności są traktowane jako encje wyższego poziomu (eliminuje redundancję)

24. [2] Co to jest pokrycie kanoniczne (4 warunki)?

Intuicyjnie kanoniczne pokrycie Fc zbioru zależności funkcyjnych F to najmniejszy zestaw zależności funkcyjnych równoważny F, w którym brak jest zbędnych zależności bądź części zależności. (czasem któreś z zależności są zbędne, a czasem tylko któreś z ich atrybutów)

Kanoniczne pokrycie dla zbioru F jest zbiorem zależności Fc takich, że:

- F logicznie implikuje wszystkie zależności w Fc
- Fc logicznie implikuje wszystkie zależności w F
- żadna z funkcyjnych zależności w Fc nie zawiera nadmiarowego atrybutu
- każda lewa strona zależności funkcyjnych w Fc jest unikalna

25. [2] Dopełnienie zbioru zależności funkcyjnych Z i jak to wyznaczyć?

- zbiór wszystkich zależności funkcyjnych logicznie wynikających ze zbioru Z nazywany jest domknięciem (dopełnieniem) zbioru Z
- Z^+ jest nadzbiorem Z
- Z^+ można znaleźć dzięki wielokrotnemu stosowaniu aksjomatów Armstronga

Algorytm

wynik := Z

powtarzaj

dla każdej zależności funkcyjnej f w wyniku:

 zastosuj zasady zwrotności i rozszerzania na f

 dodaj wynikowe zależności funkcyjne do wyniku

dla każdej pary zależności funkcyjnych f1 i f2 w wyniku:

 jeśli f1 i f2 można połączyć zasadą przechodniości

 to dodaj wynikową zależność funkcyjną do wyniku

dopóki (ostatnia iteracja nie zmieniła wyniku)

zwróć wynik (domknięcie)

26. [2] Niedogodności plików do gromadzenia danych

- Wymagana spora ilość programów do obsługi plików
- Nadmiarowość i niespójność danych
- Trudności w efektywnym dostępie do danych
- Izolacja danych
- Problemy z integralnością
- Problemy z atomowością
- Równoległy dostęp przez wielu użytkowników
- Problemy z bezpieczeństwem (logowanie, prawa dostępu)

– Trudności z optymalizacją

27. [2] REFERENCES permission, ALL permission

Uprawnienie references na tabeli daje możliwość deklarowania kluczy obcych do tej tabeli
GRANT references ON <table> TO <grupa użytkowników> (np. public - dla wszystkich)
Chcąc natomiast nadać komuś wszystkie uprawnienia używamy konstrukcji
GRANT all privileges ...

28. [2] Kto ma uprawnienia do modyfikacji schematu?

Tylko właściciel może modyfikować schemat. (NIE administrator tylko właściciel)

29. [2] Czym jest zbiór związków encji?

Zbiór związków to matematyczna relacja między dwoma lub więcej encjami. Zapisujemy ją:
(e1, e2, ..., en), gdzie każda encja ei należy do zbioru encji Ei. (tak na chłopski rozum: dla zbioru związków o stopniu 2 odpowiednikiem jest zbiór wszystkich krotek połączonych danym kluczem obcym natomiast jeden związek to jedna para krotek)
- modeluje przedsiębiorstwo jako zbiór encji i związków (zależności)
- reprezentowany schematycznie przez diagram związków encji

Encja to abstrakcyjny obiekt pewnego rodzaju. Kolekcja podobnych encji tworzy zbiór encji.
Encja ~ krotka, Zbiór encji ~ relacja (przykład 4.1 - Systemy baz danych. Kompletny podręcznik)

//encja nie jest bardziej rekordem, a zbiór encji tabelą? << zgodzę się z tym (+3)

Przecież krotka to rekord, a relacja to tabela

-1 encja to tabela a zbiór encji to tabele o tych samych atrybutach (potwierdzi ktoś?)

tutaj jest jeszcze inaczej sformułowane:

<http://math.uni.lodz.pl/~antczak/bazy/pwsz/ERD.pdf>

czytając o tym zaczynam mieć wrażenie, że są 2 podejścia do encji i zbiorów encji oraz związków i zbiorów związków... pytanie, którego ona wymaga :)

30. [2] 2 sposoby dostępu do bazy danych pod względem języków programowania

obszerniej wyjaśnione w odpowiedzi na pytanie 35

Chodzi o osadzony i dynamiczny SQL, trzeba wspomnieć, że w osadzonym zapytania są wysyłane w momencie kompilacji programu, a w dynamicznym podczas wykonywania

31. [2] Co to jest procedura składowana? Zalety?

- przechowywane w bazie danych jako część schematu
- można używać w zapytaniach SQL
- pozwalają zewnętrznym aplikacjom na operowanie na bazie danych bez znajomości wewnętrznych szczegółów
- pozwalają, aby logika biznesowa była przechowywana w bazie danych i uruchamiana z poziomu wyrażenia SQL

32. [2] Algorytm sprawdzania czy dany zbiór atrybutów jest superkluczem - wykorzystaj zbiór domknięć atrybutów

Patrz pytanie 2.

Założenia: mamy jakąś relację R o zadanych atrybutach oraz zbiór atrybutów S, który będziemy testować.

1. Wyliczamy domknięcie zbioru atrybutów S+.
2. Jeżeli S+ to wszystkie atrybuty relacji, to S jest superkluczem.

33. [1] Po co tworzymy bazy danych?

Bazy danych tworzymy po to, aby w sposób efektywny przechowywać duże dane (wyeliminować niedogodności w używaniu plików do gromadzenia danych).

Przechowywanie danych w bazie danych pozwala nam na:

- proste kierowanie zapytań i modyfikowanie danych
- przechowywanie danych przez długi czas
- zapewnienie trwałości danych
- sterowanie dostępem do danych przez wielu użytkowników jednocześnie

34. [1] Warstwy logiczne i fizyczne bazy danych

Poziom fizyczny:

- operacje na plikach rekordów
- opisuje jak rekord jest przechowywany fizycznie (np. bloki na dysku)

Poziom logiczny

- operacje na tabelach
- opisuje jakie dane są przechowywane i zależności między nimi

Poziom widoku (nie ma w pytaniu, ale jest powiązane, więc może się pojawić):

- ukrywa typy danych
- pozwala ukrywać dane z powodów bezpieczeństwa (np. poufne dane jak płace pracowników)

35. [1] Różnica między SQL dynamicznym i osadzonym

Dynamiczny SQL:

- API pozwalające programowi na współpracę z serwerem BD
- aplikacja zgłasza żądanie:
 - * połączenia z serwerem bazodanowym
 - * przesyła komendy SQL do serwera bazodanowego
 - * pobiera krotki wynikowe jedna po drugiej do zmiennych programu
- zapytania są wysyłane podczas wykonywania
- dwa standardy: JDBC i ODBC

JDBC - Javowe API do komunikacji z DBMS wspierające SQLa. Model komunikacji:

1. nawiąż połączenie.
2. stwórz obiekt "statement".
3. wykonaj zapytania używając obiektu do przesyłania zapytań oraz odbierania wyników.
4. mechanizm radzenia sobie z błędami.

ODBC - otwarty standard do komunikacji programu z serwerem bazodanowym. Model komunikacji:

1. nawiąż połączenie.
2. wyślij zapytania.
3. odbierz wyniki.

Osadzony SQL:

- Standard SQL definiuje wbudowanie SQL w różne języki programowania: C, C++, Java, Pascal, Fortran, PL/1, i inne
- język, w którym zapytania SQL są wbudowane jest określony jako język macierzysty, a struktury SQL dozwolone w języku macierzystym tworzą osadzony SQL
- wbudowany w język mechanizm wykonywania zapytań
- wyrażenie EXEC SQL (EXEC SQL <polecenie> END_EXEC choć składnia może się różnić np. EXEC-SQL w COBOLu)
- wyrażenie SQL INCLUDE SQLCA
- zanim zostaną wykonane jakiekolwiek zapytania program musi się połączyć z bazą danych (EXEC-SQL connect to SERWER user UZYKOWNIK using HASŁO)
- zmienne muszą być deklarowane w sekcji DECLARE
- żeby pisać zapytania w osadzonym SQL deklarujemy kursor, który będzie identyfikował nasze zapytania
- zapytania wysyłane w momencie kompilacji programu

36. [1] Jak się wyznacza klucz główny w zbiorach związków?

Jeśli po którejś stronie związku jest krotność 1, to wystarczy wziąć ten jeden PK. Jeśli wszędzie jest 'wiele', to kombinację wszystkich kluczy głównych wchodzących w skład związku.

37. [1] Czym są warunki integralnościowe?

Zabezpieczają bazę danych przed przypadkowym uszkodzeniem poprzez zapewnienie, że zmiany wprowadzone przez autoryzowanych użytkowników nie spowodują utraty spójności danych. Można je definiować na poziomie pojedynczego atrybutu lub całej relacji. Rodzaje:

- wartość niepusta/pusta - not null,
- klucz główny - primary key,
- unikalność danych - unique,
- zawężenie dziedziny - check
- klucz obcy - foreign key

38. [1] Integralność referencyjna

Integralność referencyjna (odwołań) - dotyczy modelowania powiązań pomiędzy obiektami. Definiujemy ją przez specyfikację klucza obcego, który może znajdować się w jednym z dwóch stanów. Może mieć wartość klucza głównego innej tabeli w bazie bądź wartość NULL (brak powiązań).

39. [1] Czy w praktyce klucz obcy może być NULLEM?

Sprzeczne opinie:

- Oczekiwana odpowiedź to zdecydowanie "NIE".
- Tak, patrz wyżej. (Przykładem może być kolumna "ReportsTo" w bazie "Northwind", gdzie "szef" miał wartość NULL)

Zgodnie z książką "Systemy baz danych. Kompletny podręcznik" (przykład 7.1) FK może mieć wartość NULL i oznacza to, że , do której się odnosimy nie istnieje (jak w przykładzie z "ReportsTo"). Może jej chodzi o to, że wartość PK w tabeli 2, który jest naszym FK w tabeli 1 nie może być NULLem? Wartość na którą FK wskazuje nie może być nullem, ale sam FK nullem może być na pewno.

40. [1] Buforowanie bloków danych

- odpowiedzialny za to jest Buffer Manager
- chodzi o to, żeby jak najwięcej bloków danych było w pamięci operacyjnej
- minimalizacja ilości przesyłanych danych z plików do bufora

41. [1] Usuwanie kaskadowe w zależnościach referencyjnych

+Przy tworzeniu tabeli można zdefiniować klucz obcy jako "on delete cascade".

W momencie usunięcia krotki, która była kluczem obcym dla innej z klauzulą "On delete cascade" ta krotka jest usuwana automatycznie.

42. [1] Transformacja atrybutów wielowartościowych, złożonych i pochodnych na model relacyjny

Atrybuty złożone rozbijamy na mniejsze, atomowe części i dla każdej tworzymy nową kolumnę. Atrybuty wielowartościowe umieszczamy w osobnej tabeli, której kolumny to klucz główny tabeli, która miała atrybut wielowartościowy oraz kolumna na nazwę atrybutu wielowartościowego. Każdą wartość wpisujemy do osobnej krotki. Atrybuty pochodne odpowiadają procedurom bezargumentowym, więc nie trzymamy ich w tabeli tylko tworzymy procedurę dla ich wyliczania. Procedurę taką można włączyć do widoku. -> mniej więcej tak odpowiedziałam i to była odpowiedź na 5.0

43. [1] Czym jest klucz kandydujący w kontekście zależności funkcyjnych?

$K \rightarrow R$, gdzie K zawarte w R (K to minimalny zbiór atrybutów potrzebny do stworzenia superklucza) a R to zbiór wszystkich atrybutów. Cytat na potwierdzenie: "Kluczem kandydującym relacji R jest zbiór jednego lub więcej atrybutów, jeśli atrybuty klucza kandydującego funkcyjnie określają wszystkie pozostałe atrybuty relacji oraz żaden podzbiór właściwy zbioru atrybutów nie określa funkcyjnie wszystkich pozostałych atrybutów R "

K jest kluczem kandydującym dla $r(R)$ wtedy i tylko wtedy gdy

- zachodzi zależność funkcyjna $K \rightarrow R$, i
- dla żadnego $\alpha \subset K$, $\alpha \rightarrow R$

44. [1] Czym jest klucz kandydujący w ER?

Minimalny zbiór atrybutów K ze zbioru R , taki że klucz złożony z atrybutów K jednoznacznie określa unikalność encji w zbiorze encji, a usunięcie dowolnego atrybutu z K usuwa jednoznaczność

45. [1] Różnice między własnymi typami, a zawężeniem dziedziny

Cieężko coś znaleźć na ten temat (slajd 77 basic_intermediate_sql daje tylko przykład, który po napisaniu w mssql BEZ słowa FINAL nie rzuca błędem).

Wydaje mi się, że może chodzić o to, że więzy check możemy nazywać dzięki czemu w dowolnym momencie można zlikwidować ograniczenie dziedziny lub je zmodyfikować*, a w przypadku własnego typu trzeba by zmienić całą jego definicję.

* modyfikacja odbywa się poprzez usunięcie i ponowne dodanie nowej definicji.

np. create constraint kwota_nieujemna check (kwota >= 0) <- w definicji tabeli lub jak niżej

alter table tabela drop constraint kwota_nieujemna

alter table tabela create constraint kwota_nieujemna check (kwota > 100)

// czy o to chodzi?

Zapytałem mniej więcej o to Zygmunt:

Nie rozumiem więc co oferuje nam definiowanie własnych typów czego nie oferuje nam zawężanie dziedziny przy użyciu więzów check/(not) null/pk/fk/unique?

np do definiowania proceduralnych rozszerzeń gdzie nie można wymusić ograniczeń (było szczegółowo omawiane na wykładzie)

46. [1] Generalizacja i specjalizacja w ER

Coś ala dziedziczenie, przykład z wykładu - zbiory encji instruktorów i studentów mogą być specjalizacją zbioru encji person.

Generalizacja działa w drugą stronę - person jest generalizacją studentów i instruktorów.

Można wyróżnić specjalizację/generalizację częściową/całkowitą - na podstawie tego, czy każdy element ze zbioru wyższego rzędu ma odpowiednik w zbiorach niższych rzędów, a także zachodzącą/rozłączną - kryterium podziału jest tutaj czy encja ze zbioru wyższego rzędu może mieć kilka specjalizacji.

47. [1] Po co są role i jak je definiujemy?

Definiujemy je poprzez "create role", służą do łatwiejszego autoryzowania użytkowników do wykonywania pewnych operacji w bazie, np. po utworzeniu roli admin można zrobić "grant <rodzaj operacji> to admin"

48. [1] Konstrukcja obsługi wyjątku

BEGIN

ciąg instrukcji do wykonania

EXCEPTION

-- Początek sekcji wyjątków

WHEN **wyjątek** THEN

ciąg instrukcji do wykonania

END;

DECLARE <gdzie przejść> HANDLER FOR <lista warunków>
<instrukcja>

Jeszcze gdzieś widziałem BEGIN TRY ... END TRY BEGIN CATCH ... END CATCH

49. [1] Prawo pierwszeństwa w harmonogramach

Wydaje mi się, że to będzie to (slajd 23 transakcje):

Dla zapewnienia szeregowalności zapobiegającej niespójności wynikającej z oddziaływań pomiędzy wykonywanymi jednocześnie transakcjami, ważne jest wzajemne uporządkowanie operacji odczytu i zapisu tych transakcji:

- jeżeli dwie transakcje jedynie czytają ten sam element danych, to ich wzajemna kolejność nie jest istotna;
- jeżeli dwie transakcje czytają lub zapisują różne elementy danych, to nie kolidują ze sobą i ich wzajemna kolejność nie jest istotna;
- jeżeli jedna transakcja zapisuje element danych, a druga go odczytuje lub zapisuje, to kolejność wykonania tych transakcji ma znaczenie.

A może to coś z grafem pierwszeństwa? (albo nie ma tego w slajdach)

Wydaje mi się, że tu chodzi właśnie o graf pierwszeństwa. To szło mniej więcej tak, że jeśli w grafie istnieje krawędź $T_i \rightarrow T_j$ to w każdym równoważnym grafie T_i występuje przed T_j , no i T_i musi wykonać się przed T_j (ale tu bym prosił potwierdzenie). Jak graf zawiera cykl to harmonogramu nie da się uszeregować.

50. [1] Metody optymistyczne/pesymistyczne tworzenia harmonogramu

metody optymistyczne:

<http://wazniak.mimuw.edu.pl/images/4/48/BD-2st-1.2-w10.tresc-1.1.pdf> slajd 16

Algorytmy optymistyczne zarządzania współbieżnością transakcji przeprowadzają transakcje przez trzy stany:

- Faza odczytu: Transakcje czytają dane z bazy danych. Wprowadzane modyfikacje są przechowywane w lokalnych obszarach roboczych transakcji
- Faza walidacji: Wykonywana jest walidacja uszeregowalności transakcji. Transakcje niespełniające kryterium uszeregowalności są wycofywane i restartowane
- Faza zapisu: Jeżeli faza walidacji zakończy się pomyślnie, modyfikacje transakcji są wprowadzane do bazy danych

Do tworzenia harmonogramu wykorzystuje się znaczniki czasowe (timestamps), które oznaczają czas wejścia transakcji w trzy fazy: Start, Validation, Finish.

Jeśli transakcja t_1 ma występować przed transakcją t_2

$\text{finish}(t_1) < \text{start}(t_2)$ albo

$\text{start}(t_2) < \text{finish}(t_1) < \text{validation}(t_2)$, przy założeniu, że t_2 nie odczytuje danych zapisywanych przez t_1

możecie też tutaj zajrzeć: <http://edu.pjwstk.edu.pl/wyklady/sbd/scb/w12.htm>

51. [1] Zarządzanie buforem

Programy wysyłają żądanie do zarządcy bufora gdy potrzebują blok z dysku

1. Jeżeli blok jest już w buforze, zarządca zwraca adres bloku w pamięci głównej
2. Jeżeli nie, to zarządca:

1. Alokuje przestrzeń w puli buforów dla bloku

1. Wymiana(wyrzucenie) jakiegoś innego bloku, w razie potrzeby, aby zrobić miejsce na nowy blok.

2. Zastąpiony blok zapisywany na dysk tylko wtedy gdy został zmodyfikowany od ostatniego czasu gdy został zapisany/pobrany na/z dysk..
2. Czyta blok z dysku do bufora i zwraca adres bloku w pamięci głównej żądającemu.

52. [1] Sandboxing

Sandbox to "bezpieczne miejsce" do wykonywania zewnętrznych skryptów. Procedury zewnętrzne wykonywane w safe access sandbox mają własną pamięć, ale ze względów bezpieczeństwa nie mają dostępu do pamięci procesów i plików BD.

53. [1] Zawężenie dziedziny, checki

Pozwalają ograniczyć wartości danego atrybutu. Można je nazywać. Umieszczamy je w definicji tabeli.

constraint <nazwa> check <atribut> <wyrażenie> (np. atrybut > 5)

Edycja poprzez usunięcie i dodanie nowej definicji.

alter table add/drop constraint <nazwa> check <atribut> <wyrażenie> (np. atrybut > 5)

54. [1] Formy autoryzacji do modyfikowania schematu bazy

Index - tworzenie i usuwanie indeksów

Resources - tworzenie nowych relacji

Alteration - dodawanie i usuwanie atrybutów w relacji

Drop - usuwanie relacji

55. [1] Legalność relacji

Legalna instancja relacji - relacja, która spełnia wszystkie nałożone zależności funkcyjne

//była jeszcze druga definicja, coś ze światem rzeczywistym - ktoś to ma? - czy ktoś może potwierdzić poniższą definicję ?

(?) relacja legalna - taka, która spełnia wszystkie wyspecyfikowane warunki integralnościowe

56. [1] Rodzaje języków

proceduralne i nieproceduralne(deklaratywne)

57. [1] Case - co robi?

W SQL case rozpatruje przypadki i w każdym przypadku należy zdefiniować warunek

case

when ... then ...

when ... then ...

else ...

end

58. [1] Algorytm wyliczania dopełnienia (ogólnie, opisowo)

$F^+ = F$

do

{

for each (zależność funkcyjna f) in (F^+)

zastosuj reguły zwrotności i rozszerzalności na f

```

    dodaj otrzymane zal. fun. do  $F^+$ 
  for each (para zal. fun.  $(f_1, f_2)$  in  $(F^+)$ 
    if  $(f_1$  i  $f_2$  mogą być połączone używając przechodniości)
      dodaj otrzymaną zal. fun. do  $F^+$ 
  }
while ( $F^+$  nie zmienia się już więcej)

```

59. [1] Diagram stanów transakcji

Każda realizowana transakcja posiada zbiór ściśle określonych stanów i zbiór ściśle określonych przejść z jednego stanu do drugiego

koniecznie NAZWAĆ konkretne stany:

- * aktywna -> częściowo wypełniona -> wypełniona->zakończona
- * aktywna -> nieudana -> zakończona
- * aktywna -> częściowo wypełniona -> nieudana -> zakończona

60. [1] Kiedy dekompozycja zachowuje zależność?

Dekompozycja R na $(R_1, R_2 \dots R_n)$ zachowuje zależność jeśli spełnia warunek

$$(F_1 \cup F_2 \cup \dots \cup F_n)^+ = F^+$$

Gdzie F_i to **restrykcja** z F nad R_i . F_i jest podzbiorem takich zależności z F^+ , które zawierają tylko atrybuty z R_i .

61. [1] Dekompozycja bezstratna + warunki

Dekompozycja jest bezstratna jeżeli nie ma utraty informacji przy zastępowaniu schematu relacji dwoma schematami relacji. Inaczej: dla wszystkich legalnych instancji bazy danych, relacja r zawiera ten sam zbiór krotek jak wynik zapytania SQL:

`select * from (select R1 from r) natural join (select R2 from r)`

Warunek: dekompozycja R do R_1 i R_2 jest złączeniem bezstratnym jeżeli przynajmniej jedna z zależności jest w F^+ :

- $R_1 \cap R_2 \rightarrow R_1$
- $R_1 \cap R_2 \rightarrow R_2$

62. [1] Czasowe zależności klucza obcego od głównego

Klucz główny powinien być stały w czasie (w MS SQL da się go zmieniać, ale nie bardzo nie powinno się tego robić) ?

// nie wiem czy to o to chodzi

Klucz obcy powinien mieć referencję do aktualnej wersji danych lub do danych w konkretnym punkcie czasowym (ostatnie 3 slajdy z wykładu o normalizacji) - może o to jej chodziło

63. [1] Harmonogramy równoważne

Różnią się kolejnością poszczególnych operacji, ale przekształcającą bazę do takiego samego stanu

64. [1] Kiedy nie używać triggerów?

- zarządzanie danymi sumarycznymi
- replikacja BD
- kaskadowe usuwanie

Ponieważ BD posiadają do trzech powyższych wbudowane mechanizmy (oraz "gdy chcemy wiedzieć co się dzieje w środku, bo nie mamy kontroli nad tym co się dzieje w triggerze" - słowa pani doktor)

65. [1] Funkcje tablicowe

create function ... returns table (...),

dają w wyniku kolekcję krotek. Mogą też przyjmować kolekcję krotek jako parametr

//coś jeszcze? <http://www.sqlpedia.pl/funkcje-tabelaryczne-sql/>

66. [1] Co to jest model danych, podaj jego elementy i przykłady modeli. Co to jest model konceptualny? Czym różni się od implementacyjnego?

Modelowanie - odwzorowanie rzeczywistych obiektów świata rzeczywistego w systemie informatycznym (bazie danych)

Model danych:

Zbiór pojęciowych narzędzi do opisu:

- danych
- zależności między danymi
- semantyki danych
- ograniczeń dla danych

definiuje: strukturę danych, możliwe operacje na danych, ograniczenia danych

- Umożliwia opisanie projektu BD na poziomie fizycznym, logicznym i widoku

Modele:

- konceptualne
- implementacyjne (modele danych)

Model konceptualny:

- reprezentacja obiektów w uniwersalnym modelu niezależnym od modelu implementacyjnego
- (związków – encji, model UML)

Model implementacyjny:

- wykorzystywany do implementacji modeli konceptualnych(relacje, obiekty, itp.)

67. [1] Funkcje składowania bazy danych

// ktoś coś ?

// może chodzi o procedury składowane?? -> są gdzieś wyżej opisane

68. [1] Definicja harmonogramu odtwarzalnego

Harmonogram odtwarzalny: Jeżeli transakcja A odczytuje wartość zapisaną poprzednio przez transakcję B to operacja wypełnienia transakcji B poprzedza operację wypełnienia transakcji A.

69. [1] Jak szeregować harmonogram

Przy założeniu o zapisie na podstawie poprzedniej wartości (transakcja modyfikuje wartość danej w oparciu o jej poprzednią wartość, którą najpierw musi odczytać), można stworzyć graf pierwszeństwa (szeregowalności), który pozwoli wykryć naruszenie szeregowalności.

Dla harmonogramu S graf pierwszeństwa to graf skierowany

$G = (N, E)$; N-zb. wierzchołków, E-zb. krawędzi skierowanych

- Jeżeli w grafie pierwszeństwa harmonogramu S istnieje krawędź $T_i \rightarrow T_j$, to w każdym harmonogramie sekwencyjnym S' równoważnym S, T_i musi występować przed T_j .
- Jeżeli graf pierwszeństwa zawiera cykl, to kolizji w harmonogramie nie można uszeregować (czyli harmonogram nie jest szeregowalny)

// ta definicja mi bardziej pasuje

Zadanie szeregowalności polega na:

rozpoznaniu harmonogramów niesekwencyjnych, które pozwalają wykonywać transakcje współbieżnie, nie dopuszczając do niepożądanych oddziaływań pomiędzy nimi i doprowadzając bazę do stanu, który mógłby równie dobrze powstać w efekcie wykonania sekwencyjnego

70. [1] Ograniczenie określające, które encje mogą być członkami danego zbioru encji w generalizacji ER

Przynależność może być:

- zdefiniowana przez ograniczenie (condition-defined)
- zdefiniowana przez użytkownika (user-defined)

//poniższe ograniczenia też dotyczą generalizacji, mogą się przydać choć nie były w pytaniu
Ograniczenie czy encje mogą należeć do więcej niż jednego zbioru encji niższego poziomu w pojedynczej generalizacji

- Rozłączne (Nie mogą zachodzić jednocześnie)
- Zachodzące (Jednocześnie)

Ograniczenie kompletności - określa czy zbiór pojęć ze zbioru encji na wyższym poziomie musi należeć do przynajmniej jednego zbioru encji na niższym poziomie

- Zupełne
- Częściowe

71. [1] Jak alternatywnie wyliczyć domknięcie zbioru zależności F+?

Przez użycie domknięcia atrybutów.

Dla każdego $\gamma \subseteq R$ znajdujemy domknięcie γ^+ i dla każdego $S \subseteq \gamma^+$ wypisujemy zależność $\gamma \rightarrow S$.

72. [1] Zależności funkcyjne

- zależność funkcyjna występuje wtedy gdy po ustaleniu wartości pewnych atrybutów relacji wartości jakichś innych atrybutów tej relacji są jednoznacznie wyznaczone (unikalne)
- notacja: $X \rightarrow Y$, gdzie X i Y są zbiorami atrybutów z relacji R
- zależność funkcyjną, której prawa strona zawiera kilka atrybutów ($X \rightarrow A_1 A_2 \dots$) można zastąpić zbiorem zależności o pojedynczych prawych stronach ($X \rightarrow A_1, X \rightarrow A_2, \dots$)
- operacja jest odwracalna
- wykorzystanie: sprawdzenie czy instancje relacji spełniają dany zbiór F zależności funkcyjnych, specyfikowanie ograniczeń na zbiorze legalnych relacji
- konkretne instancje schematu relacji mogą spełniać zależności funkcyjne nawet jeżeli zależność funkcyjna nie zachodzi na wszystkich legalnych instancjach

73. [1] Słownik danych - co to jest, co przechowuje, jak jest fizycznie zorganizowany

Słownik danych (data dictionary) przechowuje metadane, np.:

- informacje o relacjach
- informacje o użytkownikach i ich kontach, włączając hasła
- dane statystyczne i opisowe
- informacje o fizycznej organizacji plików
- informacje o indeksach

Używany zarówno przez system zarządzania bazą jak i użytkowników.

Ważny jest przedrostek w nazwie, np. user1_ Tables(Tabs) - info o tabelach, których user1 jest właścicielem

Fizyczna organizacja:

Na ogół jako mini-baza (Zbiór tabel) wewnątrz właściwej bazy danych. Żeby zapewnić szybkość dostępu często nieznormalizowana.

74. [1] W rekordzie są 3 pola o zmiennym rozmiarze atrybutów i jeden o stałym, jak to jest fizycznie zorganizowane

Atrybuty zmiennej długości reprezentowane przez stały rozmiar (offset, długość), z aktualnymi danymi przechowywanymi po wszystkich atrybutach stałej długości(słajd 12 organizacja dysku)

Dołączę się, chodziło jej dokładnie o ten rysunek, co jest na samym dole strony. (Powyższa odpowiedź nic nie dała) Oczekiwała dokładnego opisu, jak on wygląda począwszy od zerowego bajtu. Czyli najpierw lokalizacja i rozmiar każdego atrybutu, później null bitmapa, rekordy stałej długości i na końcu zmiennej.

a null bit mapa nie jest po atrybutach stałej długości? - niech ktoś odpowie :)
według rysunku jest po stałej inaczej te liczby nie miałyby sensu, chociaż kto by wyczuł czy ona się nie pomyliła

75. [1] Cechy związków encji

- Stopień związku (unarny, binarny, ternarny, n-arny...- inaczej arność związku)
- Ograniczenie krotności (1:1, 1:M, M:1, M:N)
- Istnienie: opcjonalny lub obowiązkowy

76. [1] Jak projektujemy relacyjną bazę danych

- Zebranie wymagań użytkownika i opracowanie struktury BD spełniającej te wymagania
- Wybór modelu danych i "przetłumaczenie" tych wymagań na konceptualny model BD

Proces przechodzenia z modelu konceptualnego do implementacyjnego składa się z dwóch głównych faz:

- Logiczny projekt
- Fizyczny projekt

// slajd 2 z normalizacji:

Kilka sposobów podejść, np.:

–wysokopoziomowe notacje opisu struktury danych + sposoby konwersji wysokopoziomowych projektów na relacje

–analiza wymagań + definiowanie relacji bezpośrednio (bez wykonywania etapu pośredniego w języku wykonywania etapu pośredniego w języku wysokopoziomowym)

•Niezależnie od podejścia jest miejsce na usprawnienia –zwłaszcza poprzez eliminowanie redundancji

77. [1] Buffer manager

Menedżer buforu - podsystem odpowiedzialny za alokowanie bufora do pamięci głównej. koniecznie wspomnieć, że on zapisuje z bufora na dysk

Opis działania:

Programs call on the buffer manager when they need a block from disk.

1. If the block is already in the buffer, buffer manager returns the address of the block in main memory
2. If the block is not in the buffer, the buffer manager
 1. Allocates space in the buffer for the block
 1. Replacing (throwing out) some other block, if required, to make space for the new block.
 2. Replaced block written back to disk only if it was modified since the most recent time that it was written to/fetched from the disk.
 2. Reads the block from the disk to the buffer, and returns the address of the block in main memory to requester.

78. [1] Zamiana agregacji w ER na postać relacyjną (na model relacyjny?)

//“Schematy odpowiadające agregacji”, slajd 66 w modelowaniu związków encji.

//Druga opinia potrzebna - o to chodzi w pytaniu?

Do reprezentowania agregacji, stwórz schemat zawierający:

- PK zagregowanego związku
- PK połączonego zbioru encji
- atrybuty opisowe

79. [1] Jak mają być ułożone transakcje w harmonogramie?

Z definicji harmonogramu:

Ciąg operacji wykonywanych jednocześnie transakcji, w których zachowany jest wewnętrzny porządek operacji każdej transakcji.

80. [1] Rodzaje blokad

Blokada - zmienna skojarzona z każdą daną w bazie określająca dostępność danej ze względu na możliwość wykonywania na niej określonej operacji

Blokada dzielona - transakcja może odczytywać wartość jednostki, ale nie może jej zmienić. Umożliwia innym transakcjom odczyt, ale nie modyfikację wartości.

Blokada wyłączna - transakcja może zarówno odczytywać wartość jednostki, jak i ją zmieniać. Inne transakcje nie mają możliwości odczytu ani modyfikacji.

81. [1] Co to jest rozszerzenie/redukcja blokady?

Rozszerzanie blokady - z dzielonej na wyłączną

Redukcja blokady - z wyłącznej na dzieloną

82. [1] Co to instancja? Co to jest schemat?

Instancja - stan danych (konkretne wartości w danej relacji) w OKREŚLONYM CZASIE

Schemat - logiczna struktura bazy danych

83. [1] Czym jest fizyczna izolacja danych?

Na slajdzie nazywało się to "fizyczną niezależnością danych" - chodzi o to, że dane z bazy są na dysku oddzielone od programu, nie są fizycznie jego częścią.

84. [1] Migrujące dane w widokach

// ktoś coś?

85. [1] Czym jest trigger (definicja), jak się go definiuje?

Trigger - wyrażenie wykonywane automatycznie przez system jako efekt uboczny modyfikacji BD

```
create trigger timeslot_check1
after insert on section
referencing new row as nrow
for each row
```



```
when (nrow.time_slot_id not in (
select time_slot_id
from time_slot)) /* time_slot_id not present in time_slot */
begin
rollback
end;
```

86. [1] Co to jest indeks?

Indeks - struktura danych wykorzystywana w celu przyspieszenia dostępu do rekordów o określonych wartościach dla atrybutów indeksu

```
CREATE INDEX studentID_index ON student(ID)
```

87. [1] Jak odświeżamy dane w tabeli zmaterializowanej, typy refresh'y

A nie chodzi czasem o widok zmaterializowany?

Z tego co znalazłem możliwe są 2 podejścia:

- 1) całkowite odświeżenie - czyli ponowne zmaterializowanie widoku (wyznaczenie danych i zapis na dysk). Kosztowne, ale dobre rozwiązanie jeśli dane nie muszą być aktualne. np. można dane odświeżać co noc kiedy baza nie jest używana, a analitykom wystarczą dane "z wczoraj"
- 2) przyrostowo - do widoku dodawane są nowe dane na bieżąco. Dzięki temu nie ma potrzeby materializowania widoku na nowo (od zera). Czas odświeżenia może być krótki

88. [1] Co to jest procedura skalarna?

Procedura zwracająca skalar

89. [1] O co chodzi z klauzulą with w procedurze?

Definiuje wartość, którą można użyć w procedurze.

Przykład:

```
with max_budget (value) as (select max(budget) from department)
```

90. [1] Co to jest fantom?

fantom - wiersz, który zostaje wstawiony do tabeli po tym jak transakcja wykonała operację na tej tabeli a przed jej zatwierdzeniem

91. [1] Duże pliki danych (clob/blob)

Za wiki:

-blob - typ danych, który umożliwia przechowywanie dużych ilości danych binarnych jako pojedynczy obiekt w bazie danych, stosowany w szczególności do przechowywania danych multimedialnych, takich jak grafika, muzyka czy filmy.

-clob - Dane typu CLOB różnią się od danych typu BLOB tym, iż posiadają określone kodowanie znaków.

Będzie dobrze :)

<https://www.youtube.com/watch?v=gazMgWvQPe8>

<https://youtu.be/M6wRnouGZFQ?t=21>