

Opracowanie zagadnień na egzamin z baz danych

Pytania z “eksperymentu dydaktycznego”	2
Egzamin 2017/2018	19
Pytania różne (autorskie, 2016/2017 i wcześniejsze etc.)	35

Disclaimer

Jako że slajdy Zygmunt są wewnętrznie sprzeczne oraz błędne, a ona wymaga odpowiedzi stricte na ich podstawie, to opracowanie może zawierać pewne błędy, jest jednak zgodne z jej wymaganiami i slajdami z wykładów.

Wiele pytań Zygmunt zawiera jej autorskie błędne tłumaczenia pojęć z angielskiego, przyjęliśmy tam najsensowniejszą naszym zdaniem interpretację, jednak nie ma 100% pewności, że to o to jej chodziło.

Opracował Jakub Adamczyk

Podziękowania za pomoc w stworzeniu opracowania dla: Moniki Dziedzic, Joanny Fortuny, Magdaleny Kozub, Natalii Kulczyckiej, Bartłomieja Łagosza, Mikołaja Tomalika.

Pytania z “eksperymentu dydaktycznego”

1. Co to jest relacja i jakie są jej cechy?

Relacja to zbiór rekordów o identycznej strukturze (mają takie same atrybuty o określonych typach), struktura danych przechowująca dane na temat podobnych rzeczywistych obiektów. Jest reprezentowana jako tabela składająca się z wierszy (rekordów) mających określone atrybuty (kolumny).

Cechy relacji:

- niepowtarzalne rekordy - nie może być dwóch rekordów o identycznych wartościach wszystkich atrybutów,
- brak znaczenia kolejności rekordów i atrybutów (nieuporządkowana) - są to zbiory, więc kolejność nie ma znaczenia,
- każda jest reprezentowana przez tabelę i sama reprezentuje konkretny obiekt.

2. Podaj modele inne niż relacyjny.

- Obiektowy,
- obiektowo-relacyjny,
- związków encji,
- hierarchiczny,
- sieciowy,
- nierelacyjny (NoSQL).

3. Podaj poziomy abstrakcji w systemie zarządzania bazą danych.

- 1) Fizyczny - najniższy poziom abstrakcji, opisuje sposoby, w jakie dane są faktycznie przechowywane (ich struktury danych).
- 2) Logiczny - wyższy poziom abstrakcji, mówi o tym, jakie dane są przechowywane w bazie i jakie relacje między nimi zachodzą oraz jakie warunki integralnościowe występują na danych. Opisuje bazę danych jako niewielką liczbę dość prostych struktur, których złożoność na poziomie fizycznym nie ma tutaj znaczenia (niezależność od fizycznej reprezentacji).

3) Widoku - najwyższy poziom abstrakcji, opisuje tylko wybraną część bazy danych. Zawiera informacje z części relacji, które mogą być potrzebne danemu użytkownikowi, tak, aby uprościć dla nich interakcję z bazą danych.

4. Podaj wady wieloatrybutowych pól.

- Mogą prowadzić do redundancji danych, bo te same informacje będące częścią wieloatrybutowych pól mogą być przechowywane w wielu kopiach,
- utrudniają dostęp do informacji, bo trzeba je dodatkowo wyciągać z pola,
- są trudniejsze w wykorzystaniu od jednoatrybutowych pól, np. przez powyższą konieczność wyciągania części informacji z pola.

5. Jak są reprezentowane atrybuty wielowartościowe?

Atrybuty wielowartościowe (mogące przyjmować kilka wartości, np. jeden pracownik i języki, którymi się posługuje) są przechowywane w osobnych tabelach, których kolumny to atrybuty atomowe (np. tabela EmployeeLanguages o atrybutach EmployeeID, Language).

6. Jaka jest różnica między atrybutem wielowartościowym a złożonym?

Atrybut wielowartościowy - atrybut, który może przyjmować wiele różnych wartości, przykładowo pracownik może znać wiele różnych języków.

Atrybut złożony - atrybut, który można rozłożyć na mniejsze części, nie jest atomowy, np. Address składa się z Country, City, Street.

7. Jaka jest rola kluczy?

Klucze pozwalają rozróżniać krotki w relacji. Tworzy się je na podstawie atrybutów, dlatego żadne dwie krotki nie mogą mieć wszystkich ich wartości takich samych. Są użyteczne przy definiowaniu zależności między tabelami.

8. Co to jest klucz główny, superklucz, klucz kandydujący?

Superklucz - zbiór jednego lub wielu atrybutów, które pozwalają jednoznacznie zidentyfikować krotkę (rekord) w relacji (tabeli). Jest to taki podzbiór K zbioru atrybutów R, że żadne dwie krotki z tej relacji nie mają wszystkich atrybutów R z podzbioru K identycznych. Może zawierać atrybuty nadmiarowe.

Klucz kandydujący - minimalny superklucz, który zawiera najmniejszą możliwą liczbę atrybutów. Dla danej relacji może być wiele różnych kluczy kandydujących. Jest to najmniejszy możliwy podzbiór atrybutów z podzbioru K.

Klucz główny - wybrany klucz kandydujący, który służy jako główna metoda identyfikacji krotek w relacji. Jest to cecha relacji jako całości, a nie indywidualnych rekordów.

9. Co to jest relacja walidacyjna?

Relacja walidacyjna (walidacji) to relacja przechowująca dane wykorzystywane do implementowania integralności danych. Służą np. do realizacji reguł biznesowych. Składa się zwykle z dwóch pól: klucza głównego i wartości wymaganej w jakimś polu, np. województwo i jego pełna nazwa.

10. Wymień warunki integralnościowe na pojedynczej relacji.

- 1) Not null - dany atrybut nie może przyjmować wartości null.
- 2) Unique - wartości danego atrybutu nie mogą się powtarzać (wartości w danej kolumnie muszą być unikatowe). Klucze oznaczone tym warunkiem tworzą klucz kandydujący. Dopuszcza jedną wartość null.
- 3) Primary key - oznacza dany atrybut (lub ich zbiór) jako klucz główny, automatycznie daje not null i unique.
- 4) Default - jeśli wartość atrybutu nie zostanie określona, to przyjmuje on ustaloną wartość domyślną. Maksymalnie jeden na atrybut.
- 5) Check - asercja, krotki w relacji muszą spełniać określony predykat.
- 6) Foreign key - oznacza dany atrybut jako klucz obcy.

11. Czym jest instancja bazy danych?

Instancja bazy danych to aktualna zawartość informacji w bazie danych w danym punkcie w czasie.

12. Jak przechowujemy atrybuty pochodne?

Nie przechowujemy ich, ponieważ ich wartość jesteśmy w stanie wyliczyć na podstawie wartości innego atrybutu i byłaby to niepotrzebna strata miejsca i problem przy aktualizacji danych. Dostęp do nich uzyskujemy przez stworzenie widoku.

13. Czy można przechowywać agregaty łączone w bazie danych?

Agregaty łączone, czyli wartości zwracane przez funkcje agregujące (sum, max, count) dla złączeń (grup zbiorów krotek, używając group by) nie mogą być normalnie przechowywane w bazie danych, bo to wyniki zapytań (widoki). Można za to przechowywać widok zmaterializowany realizujący to zapytanie.

14. Czym są atrybuty atomowe?

Atrybuty atomowe to takie, których wartości są elementarne, nie da się ich rozłożyć na mniejsze składowe. Opisują jedną, najbardziej podstawową cechę danego obiektu.

Atrybut jest uważany za atomowy, np. numer telefonu można uznawać za wartość atomową lub rozbić go na części reprezentujące kraj, obszar i lokalny numer, dopiero je uznając za wartości atomowe.

15. Czym jest indeks i w jakim celu się go stosuje?

Indeks to struktura danych, która ma za zadanie optymalizować wyszukiwanie rekordów w tabeli.

Wyróżnia się indeksy:

- klastrowy (klastrujący, clustered/clustering) - zakładany na maksymalnie jeden atrybut w jednej tabeli (najczęściej na klucz główny). Wskazuje fizyczny sposób zapisu rekordów w danej tabeli (kolejność ich zapisu wyznacza kolejność kluczy indeksu klastrującego).
- nieklastrowy (nieklastrujący, nonclustered/nonclustering) - może być kilka indeksów nieklastrowych na tabelę, mogą być one kompozytowe. W swojej strukturze przechowuje wskaźniki na miejsca w pamięci, gdzie znajdują się dane pola (search-key), nie zmieniają sposobu przechowywania rekordów w tabeli.
- złożone (composite) - indeks zakładany na więcej niż jednym atrybucie (np. Name + Surname).

16. W jakim celu się stosuje widoki?

- Aby ułatwić niewykształconemu informatycznie użytkownikowi bazy danych korzystanie z niej, lepiej dostosowuje bazę danych do potrzeb,
- aby ograniczyć ilość informacji udostępnianą użytkownikom, np. ukryć przed nimi część danych,
- eliminuje konieczność każdorazowego pisania zapytania, co usprawnia korzystanie z bazy danych.

17. Czym się różni widok od tabeli?

Widok (niezmaterializowany) jest przechowywany w formie zapytania SQL, podczas gdy tabela to fizyczna struktura danych. Widok może być częścią tabeli.

Główna różnica polega na tym, że dowolny widok, zmaterializowany czy nie, nie jest fragmentem logicznego schematu bazy danych, podczas gdy tabela jest.

Na tabeli można wykonywać zapytania DML (insert, delete, pod warunkiem, że mamy do tego prawa dostępowe), na widoku nie. Widok zmaterializowany można aktualizować poleceniem refresh.

Ogólnie polecenia INSERT INTO VIEW można używać na jedynie bardzo prostych widokach: klauzula FROM ma tylko jedną relację, SELECT zawiera tylko nazwy atrybutów, nie zawiera group(), having() ani agregatów czy wartości obliczonych. Wstawienie do widoku jest równoważne ze wstawieniem do tabeli, z której widok został stworzony. Atrybuty nieistniejące w widoku, a istniejące w tabeli zostaną ustawione na null.

18. Jak są przechowywane niezmaterializowane widoki?

Niezmaterializowane widoki są przechowywane w postaci zapytań. Za każdym razem, gdy chcemy zobaczyć widok, zapytanie wykonuje się na nowo.

19. Jak modyfikuje się istniejące warunki check?

Nie modyfikuje się, trzeba je usunąć i następnie stworzyć nowe. Część systemów zarządzania bazami danych udostępnia do tego celu "enable novalidate", dzięki któremu obecne w tabeli rekordy nie są sprawdzane pod kątem nowego warunku check.

20. Wymień i opisz typy reguł usuwania krotek z relacji.

- 1) Odmowy - SZBD nie usunie rekordu z tabeli nadrzędnej, tylko zachowa go i oznaczy jako nieaktywny.
- 2) Restrykcyjna - SZBD nie usunie rekordu z tabeli nadrzędnej, jeśli w tabeli podrzędnej istnieją powiązane z nią rekordy. Najpierw należy usunąć wszystkie powiązane rekordy w tabeli podrzędnej, a dopiero potem usunąć rekord z tabeli nadrzędnej.
- 3) Kaskadowa - SZBD usunie rekord z tabeli nadrzędnej, a następnie usunie wszystkie powiązane rekordy z tabeli podrzędnej oraz tablic, w których istnieje referencja do usuwanych rekordów.
- 4) Wartości NULL - SZBD usunie rekord z tabeli nadrzędnej, a następnie nada polom klucza obcego w powiązanych rekordach tabeli wartość NULL.
- 5) Wartości domyślnych - SZBD usunie rekord z tabeli nadrzędnej, a następnie nada polom klucza obcego w powiązanych rekordach tabeli bieżącą wartość domyślną podaną w specyfikacji klucza obcego.

21. Co to jest unikalny klucz klastrowy?

Unikalny klucz klastrowy to klucz wykorzystywany przez indeks klastrujący, który wyznacza także kolejność sekwencyjną w pliku wykorzystywanym do indeksowania. Jest unikalny, bo na tabeli może być założony tylko jeden indeks klastrujący.

22. Czy można zdefiniować widok przy użyciu innego widoku?

Tak, można.

23. Jaka jest różnica między indeksem klastrowym a nieklastrowym?

<u>Indeks klastrowy</u>	<u>Indeks nieklastrowy</u>
Maksymalnie 1 na tabelę	Może być kilka na tabelę
Maksymalnie na 1 kolumnę, zwykle klucz główny	Może składać się z wielu atrybutów
Jest częścią oryginalnej tabeli (nie jest osobnym obiektem)	Jest osobną strukturą danych (zwykle B ⁺ -drzewem)
Modyfikuje strukturę tabeli	Nie modyfikuje struktury tabeli

24. Czym się różni DROP [nazwa_tabeli] od DELETE FROM [nazwa_tabeli]?

DROP usuwa tabelę, a DELETE FROM usuwa zawartość tej tabeli.

25. Co tworzy superklucz dla zbioru związków encji?

Kombinacja kluczy głównych zbiorów encji biorących udział w związku.

26. Czym jest zbiór słabych encji i czym różni się od zwykłego zbioru encji?

Zbiór słabych encji to zbiór encji bez klucza głównego (w przeciwieństwie do zwykłego zbioru encji), kluczem głównym słabej encji jest połączenie klucza głównego encji identyfikującej (z którą słaba encja jest w związku) oraz dyskryminatora (atrybut lub zbiór atrybutów ze zbioru słabych encji, który pozwala na rozróżnienie encji z tego zbioru).

27. Czym różni się generalizacja od specjalizacji w modelowaniu ER?

Są równoważne, różnią się tylko sposobem podejścia (bottom-up: generalizacja, top-down: specjalizacja).

28. Co to jest dyskryminator zbioru słabych encji?

Dyskryminator to atrybut lub zbiór atrybutów w zbiorze słabych encji, który wraz z kluczem głównym encji identyfikującej tworzy klucz główny słabej encji, jest podkreślony przerywaną linią na diagramie.

Przykładowo: w Conference_Days mamy DayNumber, nie unique, ale jednoznacznie określa dni należące do tej samej konferencji. DayNumber- dyskryminator, (DayNumber,ConferenceNumber) - klucz główny

29. Ilu minimalnie zbiorów encji wymaga związek wiele-do-wiele?

1, gdyż może wystąpić związek rekurencyjne zbioru samego ze sobą, który wymaga tylko tego zbioru i jeden z jego atrybutów pełni 2 role (przy czym dla analogicznej sytuacji z tabelami trzeba by już użyć 2 tabel - dodatkowej przejściowej).

30. Czy każdy związek n-arny można przekształcić na binarne?

Tak, ale przestrzegając reguł dla zachowania jednoznaczności. Tworzymy sztuczny zbiór encji E, atrybuty z R dodajemy do E.

31. Wymień etapy projektowania bazy danych i krótko je scharakteryzuj.

- 1) Analiza wymagań, definiuje się cel i założenia wstępne, tworzy się tekstowy opis wymagań.
- 2) Projekt konceptualny, tworzy się opis bazy danych na wysokim poziomie abstrakcji (model ER).
- 3) Projekt logiczny, tworzy się relacyjny model bazy danych (tabele, atrybuty).
- 4) Projekt fizyczny, tworzy się struktury wykorzystywane podczas faktycznego przechowywania bazy danych.

32. Czym są role w związkach encji?

Rola charakterystyka opisująca uczestnictwo encji w powiązaniu. Służy do opisanie natury tego uczestnictwa, jak również umożliwia rozróżnienie wielokrotnego uczestnictwa encji w tym samym powiązaniu, np. osoba (encja) może mieć rolę męża lub żony w powiązaniu “małżeństwo”.

33. Co dziedziczy zbiór encji od zbioru encji wyższego poziomu?

Dziedziczy wszystkie atrybuty oraz udziały w związkach encji.

34. Czym jest agregacja?

Agregacja to rodzaj abstrakcji, w którym zbiory encji i związki między nimi traktuje się jak encje wyższego poziomu. Pozwala to np. tworzyć związki takiej encji wyższego poziomu z innymi encjami (zwykłymi lub podobnie stworzonymi). Eliminuje to pewne ograniczenia modelu ER.

35. Jakie są rodzaje pytań do identyfikacji zależności?

- Skojarzeniowe (asocjacyjne) - czy pojedynczy rekord z danej tabeli może zostać skojarzony z 1 lub N rekordami z drugiej tabeli?
- Kontekstowe - czy rekord z jednej tabeli zawiera/ma/stanowi część/posiada (zorientowane na własność)?

Co robi/Kogo odwiedza/umieszcza/naucza (zorientowane na działanie)?

36. Z jakiej tabeli definiujemy regułę usuwania i dlaczego?

Z tabeli nadrzędnej (w stosunku do tej, z której chcemy usunąć dane), bo tam widać wszystkie możliwe problemy, a samą regułę usuwania piszemy w tabeli podrzędnej (bo tam jest FK i do niego dopisuje się regułę usuwania).

Regułę usuwania piszemy zawsze w tabeli podrzędnej jako dodatkową opcję do klucza obcego:

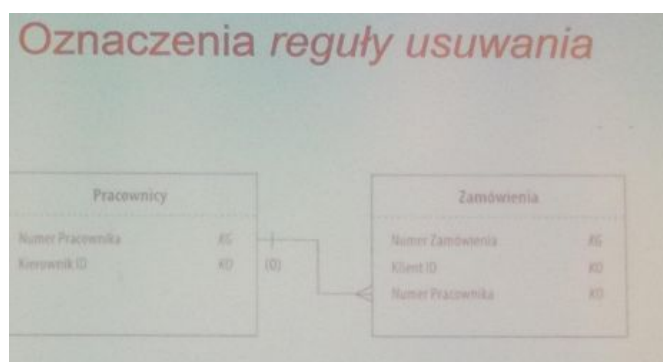
CONSTRAINT fk_name

FOREIGN KEY (child_col1, child_col2, ... child_col_n)

REFERENCES parent_table (parent_col1, parent_col2, ... parent_col_n)

ON DELETE CASCADE

Reguły te określają, co powinien zrobić SZBD, gdy pojawi się żądanie usunięcia rekordu z tabeli nadrzędnej. Zapobiega to powstawaniu osieroconych/wiszących rekordów. Na schemacie oznacza się je po stronie tabeli nadrzędnej, przykład (z regułą odmowy):



37. Narysuj przykład zbioru słabych encji połączonych ze zbiorem identyfikującym encji za pomocą związku identyfikującego.

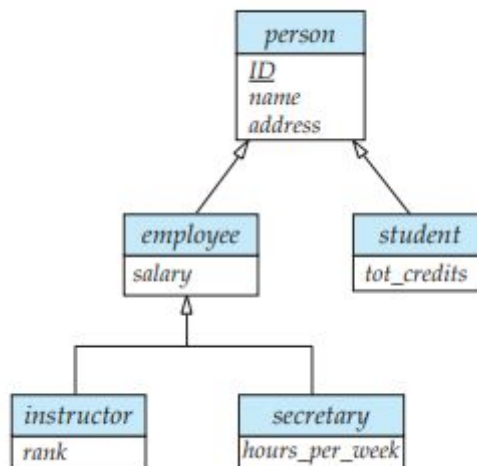
Słabe encje to szczególny typ encji, które nie mają klucza głównego (identyfikuje się je dyskryminatorem, patrz pytanie 30). Związki identyfikujące, które wskazują zbiory encji, których klucze wykorzystywane są przez zbiory słabych encji, reprezentowane są za pomocą podwójnego równoległoboku:



38. Opisz proces projektowania bottom-up i top-down.

Bottom-up - generalizacja, proces projektowania od szczegółu do ogółu.

Top-down - specjalizacja, proces projektowania od ogółu do szczegółu.



Jak mógł być zaprojektowany powyższy diagram:

- bottom-up: wiemy, że będziemy trzymać informacje o studentach i pracownikach. Mają oni niektóre atrybuty wspólne (ID, name, address), ale też się między sobą różnią. Należy więc stworzyć tabelę person, która jest generalizacją student i employee.
- top-down: wiemy, że w bazie będziemy trzymać informacje o osobach, stwórzmy więc tabelę person. Będą wśród nich studenci i pracownicy -

stwórzmy tabele student i employee, które będą dziedziczyć podstawowe informacje z person i będą miały również swoje indywidualne atrybuty. Z obu podejść powstają te same schematy.

39. W jaki sposób przedstawia się graficznie atrybut zbioru związków? Opisz lub narysuj przykład.

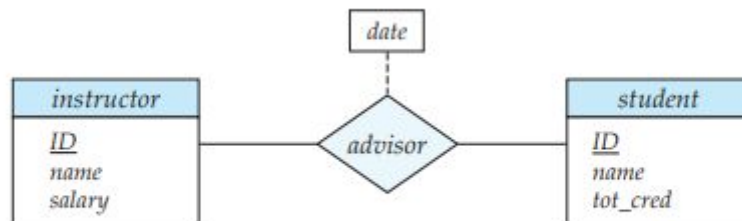


Figure 7.8 E-R diagram with an attribute attached to a relationship set.

Od symbolu zbioru związków rysuje się linię przerywaną do prostokąta, w którym umieszczamy nazwę atrybutu.

40. Dlaczego dopuszcza się maksymalnie jedną strzałkę poza związek ternarny?

Gdyż przy wielu strzałkach wystąpiłaby niejednoznaczność.

41. Czym jest funkcja agregująca w algebrze relacji i jak jest oznaczana? Wymień kilka przykładowych funkcji.

Funkcja agregująca bierze zbiór wartości i zwraca w wyniku jedną wartość, np. avg(), count(), max(), min(), sum(). Jest oznaczana przez G pisane, np. $G_{sum(a)}(r)$ dla sumowania atrybutu a w relacji r .

42. Czym jest stopień relacji?

Stopień relacji to liczba atrybutów relacji.

43. Wymień i krótko opisz operatory rozszerzonej algebry relacji.

- Uogólniona projekcja - rozszerza operację projekcji, tak, że można używać operatorów arytmetycznych w liście projekcji,
- Funkcja agregująca - przyjmuje zbiór wartości i zwraca pojedynczą wartość.

44. Wymień 6 podstawowych operatorów algebry relacji.

- 1) Selekcja (σ) - wybiera krotki spełniające predykat, np.:

SELECT * from address WHERE cityID = 5

- 2) Projekcja (Π) - wybiera atrybuty relacji, np.:

SELECT city, street FROM address

- 3) Unia (\cup) - łączy dwie lub więcej relacji w jedną:

$$A \cup B = \{x : x \in A \text{ or } x \in B\}$$

- 4) Różnica ($-$) - wybiera krotki z pierwszej relacji, które nie występują w drugiej relacji, np.:

SELECT city FROM tabela1 EXCEPT SELECT city FROM tabela2

- 5) Produkt kartezjański (\times) - tworzy iloczyn kartezjański dwóch relacji, np.:

SELECT * FROM tabela1 CROSS JOIN tabela2

- 6) Przemianowanie (ρ) - nie powoduje żadnych zmian w zawartości relacji, służy jedynie do zmiany nazwy relacji i/lub zmiany nazwy jej atrybutów.

45. Jakie korzyści daje nam operator przemianowania?

- Pozwala na nazwanie wyników wyrażeń algebry relacji (na odnoszenie się do nich),
- pozwala odnosić się do relacji za pomocą więcej niż jednej nazwy.

46. Co to znaczy, że dziedziny atrybutów są kompatybilne?

Kompatybilne dziedziny to takie, których zbiór przyjmowanych wartości jest taki sam.

47. Czym się różni złączenie wewnętrzne od zewnętrznego?

Złączenie zewnętrzne zapobiega utracie danych poprzez dodanie do wyniku krotek z pierwszej relacji, które nie pasują do krotek w drugiej relacji używając wartości null.

48. Czym jest uogólniona projekcja?

Uogólniona projekcja rozszerza operację projekcji zezwalając, aby operatory matematyczne były użyte w liście projekcji, np. w relacji z roczną wypłatą dla pracownika można uogólnioną projekcją dostać miesięczną wypłatę, bo można zrobić salary/12.

49. Co nam daje operacja przypisania w algebrze relacji?

Pozwala wygodnie wyrażać złożone zapytania, przedstawiać je jako program sekwencyjny składający się z przypisań i wyrażenia zwracającego wynik zapytania.

50. Czym charakteryzuje się czysta algebra relacji i czym różni się od algebry wielozbiorów?

Usuwa wszystkie duplikaty, jeśli występują takie w wyniku, podczas gdy algebra wielozbiorów pozostawia duplikaty.

51. Jakie warunki muszą być spełnione aby można było zastosować operator unii?

- Łączone relacje muszą mieć taki sam stopień,
- dziedziny atrybutów łączonych relacji muszą być kompatybilne.

52. Przetłumacz podzapytanie algebry relacji na podzapytanie w SQL.

$$\prod_{OrderID, ShipVia} (\sigma_{[Order\ Details].OrderID = Orders.OrderID} (\sigma_{ProductID=1} Orders \times [Order\ Details]))$$

SELECT OrderID, ShipVia

FROM OrderDetails CROSS JOIN Orders

WHERE [Order Details].OrderID = Orders.OrderID AND ProductID = 1

53. Czym jest grupa “czystych” języków?

Są to języki deklaratywne o takiej samej mocy, co algebra relacji (z tego powodu również formalne), np. wnioskowanie na krotkach czy wnioskowanie dziedzinowe.

54. Czym jest złączenie naturalne?

Złączenie naturalne bierze sumę relacji R i S, a następnie bierze pary krotek t_r z R i t_s z S i jeśli mają taką samą wartość na każdym atrybucie z przecięcia zbiorów R i S, to dodaje krotkę do wyniku.

55. Jakie są kategorie języków podzapytań?

- Proceduralne, np. algebra relacji,
- Deklaratywne, np. wnioskowanie na krotkach, wnioskowanie dziedzinowe.

56. Wymień i opisz postaci normalne.

1NF - tabele muszą zawierać tylko atrybuty atomowe (nie mogą zawierać atrybutów złożonych, wielowartościowych ani relacji zagnieżdżonych).

2NF - tabela jest w 1NF oraz każdy niekluczowy atrybut relacji musi być w pełni funkcyjnie zależny od całego klucza kandydującego.

3NF - tabela jest w 2NF oraz wszystkie atrybuty tabeli są zależne tylko od kluczy kandydujących.

Definicja alternatywna: jeśli dla wszystkich zależności funkcyjnych $A \rightarrow B$ zachodzi: $A \rightarrow B$ jest trywialne (nie może nie zachodzić, zbiór B zawiera się w A) lub A jest superkluczem lub B jest atrybutem kluczowym.

BCNF - jeśli dla wszystkich zależności funkcyjnych $A \rightarrow B$ zachodzi: A jest superkluczem lub $A \rightarrow B$ jest trywialne.

"Data depends on the key [1NF], the whole key [2NF] and nothing but the key [BCNF]."

"Non-key data depends on the keys and nothing but the keys [3NF]."

57. Wymień korzyści płynące z zastosowania postaci normalnych.

- Nie występuje redundancja danych,
- eliminacja trzech rodzajów anomalii (wstawiania, modyfikacji i usuwania),
- lepsza organizacja struktury bazy danych,
- bardziej spójne dane.

58. Co to jest dekompozycja stratna?

Dekompozycja stratna to taka dekompozycja (rozbicie tabeli na wiele mniejszych), że później nie można odtworzyć z nich oryginalnej tabeli. Innymi słowy po ich połączeniu (join) tworzą się błędne, nadmiarowe rekordy, których nie było w oryginalnej tabeli. "Strata" z nazwy dotyczy utraty informacji o prawdziwych rekordach, nie samej liczby rekordów po dekompozycji i połączeniu z powrotem.

59. Wymień własności dekompozycji.

Prawidłowa dekompozycja:

- jest bezstratna (nie powoduje utraty krotek z oryginalnej tabeli, po join'ie tabel powstanie z powrotem oryginalna tabela),
- zachowuje zależności,
- nie powoduje redundancji danych.

60. Jakie problemy eliminuje pierwsza postać normalna?

- Zabrania atrybutów wielowartościowych, złożonych i ich kombinacji,
- zmniejsza redundancję danych,
- eliminuje relacje zagnieżdżone (wielowartościowe atrybuty, które są również złożone).

61. Czym są zależności funkcyjne?

Zależność funkcyjna to constraint pomiędzy dwoma zbiorami atrybutów, który wynika ze znaczenia i wzajemnych powiązań między atrybutami. Jest to wymaganie, aby wartość pewnego zbioru atrybutów określała wartość innego zbioru atrybutów. Działa jak funkcja, bo odwzorowuje jednoznacznie argument na wartość.

Formalnie zależność funkcyjna $A \rightarrow B$ między dwoma zbiorami atrybutów A i B z relacji R wtedy i tylko wtedy, gdy każde dwie krotki t_1 i t_2 zgodne w A są zgodne w B , tzn. $t_1[A] = t_2[A] \Rightarrow t_1[B] = t_2[B]$.

Występują trzy typy zależności funkcyjnych postaci $A_1 A_2 \dots A_n \rightarrow B_1 B_2 \dots B_m$:

- trywialne - jeśli zbiór atrybutów $B_1 B_2 \dots B_m$ jest podzbiorem zbioru atrybutów $A_1 A_2 \dots A_n$, zależność zawsze zachodzi,
- nietrywialne - jeśli co najmniej jeden atrybut B_i nie należy do zbioru atrybutów $A_1 A_2 \dots A_n$,
- całkowicie nietrywialne - jeśli żaden z atrybutów B_i nie należy do zbioru atrybutów $A_1 A_2 \dots A_n$.

62. Wymień warunki, jakie musi dodatkowo spełniać relacja w 3NF, aby spełniać BCNF.

Dla każdej zależności funkcyjnej $A \rightarrow B$ zbiór A musi być superkluczem lub $A \rightarrow B$ jest trywialna.

63. Co to znaczy, że dziedzina atrybutu jest atomowa?

Atomowość oznacza, że atrybut zawiera dokładnie jedną informację, np. rok urodzenia. Przykładem nieatomowej dziedziny jest adres, np. Kraków, ul. Kijowska 57, bo zawiera 3 informacje (miasto, ulica, numer domu).

64. Na czym polega proces normalizacji?

Normalizacja schematu bazy danych polega na jego przekształceniu do postaci, w której nie będą występowały następujące anomalie (postaci normalnej):

- redundancja danych,
- anomalie wstawiania - wstawienie pewnych atrybutów do tabeli nie jest możliwe bez obecności innych atrybutów (przez co np. wstawienie prawidłowego rekordu jest niemożliwe),
- anomalie modyfikacji - zmodyfikowanie wartości atrybutu jednej krotki może wymagać zmodyfikowania wartości atrybutów innych krotek,
- anomalie usuwania - usunięcie zbędnej informacji może prowadzić do usunięcia informacji przydatnych.

Prościej: normalizacja to proces testowania relacji pod względem spełniania warunków postaci normalnych.

65. Co to jest atrybut główny?

Atrybut główny (kluczowy) to atrybut należący do jakiegoś klucza kandydującego.

66. Co można zrobić z atrybutami, które często przyjmują wartość NULL?

- Rozbić relację przy użyciu dekompozycji i stworzyć związek między tabelami, brak powiązania będzie oznaczał wartość NULL danego atrybutu (używając klucza obcego),
- rozbić relację na wiele z osobnymi kluczami głównymi, np. rozdzielić osoby na te, o których mamy informację o miejscu zamieszkania i te, o których nie mamy.

67. Jakie są dozwolone interpretacje wartości NULL?

- Atrybut nie ma zastosowania do tej krotki,
- wartość atrybutu jest nieznana,
- wartość będzie znana, ale jeszcze jest niedostępna/nie zapisana.

68. Czym jest dekompozycja?

Dekompozycja relacji to rozbięcie jednej relacji na dwie lub więcej. Kiedy dekomponujemy relację $A (A_1, A_2, A_3, \dots, A_n)$ na relacje $B (B_1, B_2, B_3, \dots, B_m)$ i $C (C_1, C_2, C_3, \dots, C_r)$, muszą być zachowane następujące warunki:

- $A = B \cup C$, czyli B i C muszą zawierać wszystkie elementy z A (nie można tracić informacji),
- krotki w relacji B i C to krotki powstałe przez rzutowanie krotek relacji A na zestaw atrybutów $(B_1, B_2, B_3, \dots, B_m)$ i $(C_1, C_2, C_3, \dots, C_r)$.

69. Wymień trzy reguły Armstronga.

Niech $R(U)$ będzie schematem relacji opisanym na zbiorze atrybutów U . Symbole X , Y , Z niech oznaczają dowolne podzbiory U , XY oznacza sumę zbiorów X i Y .

Aksjomat zwrotności [edytuj | edytuj kod]

$$Y \subseteq X \Rightarrow X \rightarrow Y$$

Aksjomat powiększenia [edytuj | edytuj kod]

$$X \rightarrow Y \Rightarrow XZ \rightarrow YZ$$

Aksjomat przechodności [edytuj | edytuj kod]

$$X \rightarrow Y \wedge Y \rightarrow Z \Rightarrow X \rightarrow Z$$

70. Co to jest denormalizacja?

Denormalizacja to proces przekształcania i przechowywania złączenia relacji wyższych postaci normalnych jako relacji bazowej w niższej postaci normalnej ze względów wydajnościowych. Można powiedzieć, że jest to wprowadzanie kontrolowanej redundancji w celu poprawienia wydajności.

Egzamin 2017/2018

Tura I

1. Co znaczy wartości NULL w krotkach i dlaczego się jej nie powinno nadużywać?

Oznacza wartość nieznaną/nieokreśloną, do późniejszego wstawienia lub niemającą zastosowania dla danej krotki. Nie należy jej nadużywać, gdyż marnuje miejsce, powoduje problemy przy złączeniach i tworzeniu agregatów.

2. Jakie niechciane zależności pozwala nam wyeliminować przejście do 2NF?

Eliminuje ona krotki zależne funkcyjnie od tylko części klucza kandydującego (2NF wymaga zależności od całego klucza).

3. Co to jest DDL i do czego służy?

DDL (Data Definition Language) to język używany do tworzenia i modyfikacji struktur bazy danych oraz ich własności, czyli np. tworzenie i usuwanie tabel/indeksów, dodawanie kolumn.

4. Czym różni się superklucz od klucza kandydującego?

Superklucz to zbiór atrybutów, który pozwala jednoznacznie identyfikować krotki w relacji, a klucz kandydujący to superklucz o najmniejszej możliwej liczbie atrybutów (minimalny podzbiór atrybutów superklucza).

5. Wymień różnice między zbiorem encji i zbiorem związków. Które z nich może posiadać atrybuty?

Zbiór encji składa się z encji tego samego typu (obiektów, np. ludzi, adresów), a zbiór związków z relacji pomiędzy encjami (np. zbiór połączeń pomiędzy ludźmi a ich adresami zamieszkania). Oba mogą posiadać atrybuty.

6. Do czego stosuje się relację walidacji?

Do sprawdzania integralności danych, do realizacji reguł biznesowych.

7. Co robi kod “DROP VIEW zarobki RESTRICT”?

Usunie widok “zarobki”, jeśli nie zależą od niego jakiegokolwiek inne obiekty w bazie. W przeciwnym wypadku wyrzuci błąd i niczego nie usunie.

8. Z czego składa się model danych?

Ze zbioru narzędzi konceptualnych do opisu danych, relacji między nimi, informacji semantycznych (znaczenia danych) oraz warunków (ograniczeń) integralnościowych.

9. Co to jest modelowanie i czym się różni modelowanie konceptualne od implementacyjnego?

Jest to odwzorowanie obiektów świata rzeczywistego w systemie informatycznym. Modelowanie konceptualne to reprezentacja obiektów w uniwersalnym modelu niezależnym od implementacji (np. ER, UML), a implementacyjne to tworzenie modeli do implementacji modelu konceptualnego (np. relacyjny, obiektowy).

10. Co to jest indeks klastrowy?

Jest to indeks wyznaczający faktyczną fizyczną kolejność rekordów w tabeli, max 1 na tabelę i max na 1 kolumnę, zwykle klucz główny.

11. Czym różni się algebra relacji od algebry wielozbiorów?

Algebra relacji nie pozwala na powtórzenia w wynikach (usuwa duplikaty), a algebra wielozbiorów na to pozwala.

12. Opisz, co robi

$$\Pi_{name, course_id}(\sigma_{instructors.ID=courses.ID}(\sigma_{dept_name='physics'}(instructors) \times courses))$$

Wybiera imię instruktora i numer prowadzonego przez niego kursu z wydziału fizyki.

```
SELECT name, course_id
```

```
FROM instructors CROSS JOIN courses
```

```
WHERE dept_name = "physics" AND instructors.ID = courses.ID
```

13. Wymień korzyści fizycznej izolacji danych.

Można dzięki temu łatwo zmienić fizyczną reprezentację danych (np. na wydajniejszą) bez wpływu na jej schemat logiczny.

14. Co to jest słownik danych i w jaki sposób jest na ogół realizowany?

Jest to specjalna tabela zawierająca metadane zwracane przez DDL zawierające informacje o danych z bazy danych i o jej strukturze (np. jej schemat), korzysta z niej głównie system zarządzania bazą danych. Słownik danych dla całej bazy danych realizuje się jako miniaturową bazę danych wewnątrz właściwej bazy, zbiór tabel (często nieznormalizowanych dla wydajności).

15. Czym jest dekompozycja stratna i jak sobie z nią radzić?

Jest to dekompozycja, podczas której pojawiają się błędne krotki (tracona jest informacja o oryginalnej tabeli) lub nie są przestrzegane zależności funkcyjne. Aby sobie z nią poradzić należy normalizować tabele oraz wykonywać tylko takie dekompozycje, które zachowują zależności funkcyjne.

16. CREATE VIEW widok(id,total_coś) AS (SELECT ..., SUM(coś) FROM ... GROUP BY ...) - czy taki widok można modyfikować? Uzasadnij.

Nie można, bo modyfikować można tylko widoki bezpośrednio używające kolumn, a nie wartości wyliczanych z nich, czyli np. agregatów (jak sum w przykładzie) czy działań na zbiorach (np. union, except).

17. Co to są role?

Role to stanowiska związane z nadanymi im uprawnieniami w bazie danych. Użytkownicy mają nadane role i dzięki temu odpowiednie autoryzacje. Role można przekazywać użytkownikom lub innym rolom (dziedziczą swoje uprawnienia).

18. Czym jest zależność funkcyjna? Dlaczego system nie może automatycznie określić tych zależności? Czy zawsze można (albo czemu nie można) określić zależności funkcyjnych na podstawie obecnego stanu relacji?

Zależność funkcyjna to constraint pomiędzy 2 zbiorami atrybutów wynikający z ich znaczenia i powiązań. Jest to wymaganie, aby wartość jednego zbioru atrybutów określała wartość drugiego zbioru.

Nie można ich automatycznie określić, bo nie mogą być wnioskowane z instancji, muszą być określone wprost.

Nie można ich określić na podstawie obecnego stanu relacji, bo zależność to własność schematu relacji, a nie jej instancji.

19. Kiedy dekompozycja zachowuje zależności funkcyjne?

Kiedy dla wszystkich tabel otrzymanych po dekompozycji zachodzą odpowiednie zależności funkcyjne z pierwotnej tabeli (kiedy żadne zależności nie przestają zachodzić po dekompozycji).

20. Co to jest jednoznaczna semantyka atrybutów relacji?

Oznacza, że znaczenie atrybutów (ich semantyka) powinno być jednoznaczne, bezsprzecznie określać, o czym mówi dany atrybut.

21. Jak wygląda struktura słownika danych?

Jest to struktura miniaturowej bazy danych - tabele i połączenia między nimi, zwykle są zdenormalizowane dla wydajności.

Tura II

1. Jakie są komponenty systemu zarządzania bazą danych?

Przestrzeń dyskowa przechowująca dane, menadżer składowania (zawiera menadżerów buforowania, plików, integralności i transakcji), procesor kwerend (silnik ewaluacji, kompilator, linker, interpreter), narzędzia administracji, tworzenia kwerend i interakcji z bazą danych.

2. Jak jest wykonywane zapytanie?

- 1) Parsowanie i translacja,
- 2) Optymalizacja
- 3) Ewaluacja

3. Wymień i opisz typy integralności danych.

Domenowa (dziedzinowa) - każdy atrybut ma dane z określonej dziedziny, np. ciągi znaków lub liczby typu Integer.

Referencyjna - dane nawiązujące do innych tabel (klucze obce) muszą być integralne pomiędzy tabelami (mieć takie same odpowiednie wartości).

Asercje - dane muszą spełniać określone predykaty (asercje) realizujące opisywane przez nie ograniczenia integralnościowe.

Autoryzacja - dane mogą być modyfikowane (wstawiane, usuwane, zmieniane) tylko przez użytkowników z określonymi uprawnieniami, dzięki czemu zwykły użytkownik nie może doprowadzić do utraty integralności danych, do których nie ma dostępu.

//Powyższa wersja zakłada, że chodzi ogólnie o integralność danych, jeśli chodzi o pojedynczą tabelę (integralność 1 danej), to patrz odpowiedź poniżej

Not null - dany atrybut nie może przyjmować wartości null.

Unique - wartości danego atrybutu nie mogą się powtarzać (wartości w danej kolumnie muszą być unikatowe). Klucze oznaczone tym warunkiem tworzą klucz kandydujący. Dopuszcza jedną wartość null.

Primary key - oznacza dany atrybut (lub ich zbiór) jako klucz główny, automatycznie daje not null i unique.

Default - jeśli wartość atrybutu nie zostanie określona, to przyjmuje on ustaloną wartość domyślną. Maksymalnie jeden na atrybut.

Check - asercja, krotki w relacji muszą spełniać określony predykat.

4. Kiedy można pominąć warunki integralnościowe?

Kiedy kluczowa jest wydajność, a wstawiana do bazy duża ilość danych na pewno spełnia te warunki (wiemy to z jakiegoś zewnętrznego źródła), lub kiedy dodajemy constraint i nie chcemy sprawdzać starych danych (bo np. wiemy, że go nie spełniają).

5. Jaki może być udział zbioru encji w zbiorze związków?

Całościowy (wszystkie encje muszą brać udział w związku ze zbioru) lub częściowy (tylko część encji bierze udział w związku).

6. Jak są przechowywane atrybuty pochodne?

Nie są przechowywane, są wyliczane za każdym razem, kiedy są potrzebne.

7. Co robi

$$\Pi_{course_id}(\sigma_{Semester="Fall" \text{ and } year(date)=2009}(Courses)) \cap \Pi_{course_id}(\sigma_{Semester="Spring" \text{ and } year(date)=2010}(Courses))$$

Wybiera course_ID kursów, które odbywały się zarówno jesienią 2009 roku, jak i wiosną 2010 roku.

```
SELECT course_ID
```

```
FROM Courses
```

```
WHERE Semester = 'Fall' AND YEAR(date) = 2009
```

```
INTERSECT
```

```
SELECT course_ID
```

```
FROM Courses
```

```
WHERE Semester = 'Spring' AND YEAR(date) = 2010
```


8. Jakich sytuacji zabrania 1NF? Wymień kilka.

Zabrania występowania atrybutów wielowartościowych i złożonych (nieatomowych) oraz relacji zagnieżdżonych. Przykład: pole Adres składające się z kraju, miasta, ulicy i numeru domu; pole Pracownik składające się z jego imienia, nazwiska i ID.

9. Jakich zależności zabrania 3NF?

Zabrania zależności funkcyjnych atrybutów tabeli od atrybutów niekluczowych.

10. Czy widok SQL o danej definicji może być używany do wprowadzania modyfikacji (dany widok z joinem)? Uzasadnij.

Nie może, bo w widokach korzystających z wielu tabel poprzez join'y można modyfikować tylko jedną wykorzystywaną w nim tabelę.

11. Co robi WITH CHECK w definicji widoku?

Wymusza na wszystkich modyfikacjach danych wykonywanych na widoku przestrzeganie warunków zdefiniowanych w klauzuli WHERE z SELECT'a tworzącego widok.

12. Opisz dekompozycję schematu do BCNF.

Najpierw należy znormalizować relację R do 3NF (dziedziny wszystkich atrybutów atomowe, wszystkie atrybuty niekluczowe zależne tylko od kluczy kandydujących), a następnie dla każdej nietrywialnej zależności funkcyjnej $A \rightarrow B$ zdekomponować relację R do relacji $(A \text{ suma } B)$ oraz $(R - (B - A))$.

13. Co robi "DROP VIEW nazwa CASCADE"?

Usuwa widok oraz wszystkie obiekty wykorzystujące go (np. inne widoki).

14. Co to jest indeks nieklastrowy?

Jest to indeks niewpływający na fizyczną kolejność rekordów w tabeli, zakładany dla większej wydajności wyszukiwania. Może być ich wiele na tabelę, mogą być zakładane na wielu atrybutach, są przechowywane w postaci wskaźników.

15. Kiedy robimy klastrowanie wielotablicowe?

Kiedy przechowujemy rekordy wielu różnych relacji powiązanych ze sobą (np. często razem wykorzystywanych) w jednym pliku oraz kiedy przechowujemy te rekordy na jednym bloku. Przyspiesza to operacje wejścia/wyjścia na tych rekordach dla wszystkich tabel, z których one pochodzą.

16. Czym jest hurtownia danych i czemu służy?

Hurtownia danych to specjalistyczna baza danych służąca do gromadzenia danych historycznych z wielu baz danych do ich analizy pod kątem biznesowym (np. użyteczności marketingowej czy maksymalizacji sprzedaży).

17. Omówić przechowywanie rekordów o atrybutach stałej i zmiennej długości.

Dla rekordów z atrybutami o stałej długości alokuje się dla każdego tyle miejsca, ile miejsca on zajmuje (dzięki stałej długości od razu to wiadomo), a następnie na bloku pamięci alokuje się miejsce dla jak największej całkowitej liczby rekordów (może przez to zostać trochę wolnego miejsca).

Dla rekordów z atrybutami o zmiennej długości: najpierw przechowuje się pary (*offset, length*), gdzie *offset* to miejsce w pliku, gdzie zaczyna się atrybut i *length* to długość atrybutu w bajtach, potem alokuje się pamięć dla atrybutów o stałej długości, a następnie null bitmapę reprezentującą wartości null (jeśli bit jest 1, to odpowiednia wartość jest null) i na koniec atrybuty o zmiennej długości.

18. Jak może skończyć się wykonywanie transakcji?

Sukcesem (odpowiednie działania na bazie danych są wykonywane normalnie) lub porażką (w bazie danych nie są dokonywane żadne zmiany, tak, jakby transakcja nigdy nie miała miejsca).

19. Wymień i krótko opisz 3 rodzaje anomalii.

- 1) Wstawiania - kiedy nie można wstawić prawidłowej krotki bez wcześniej obecności w tabeli określonych wartości atrybutów
- 2) Modyfikacji - zmiana wartości atrybutu jednej krotki wymaga modyfikacji atrybutów innych krotek
- 3) Usuwania - usunięcie zbędnych informacji pociąga za sobą usunięcie przydatnych informacji

20. Kiedy dana zależność funkcyjna jest trywialna?

Kiedy zachodzi zawsze (nie może nie zachodzić), tzn. jest postaci $A \rightarrow B$ i zbiór B zawiera się w zbiorze A.

21. Czym są zasady wnioskowania na zależnościach funkcyjnych?

Są to zasady pozwalające wnioskować o istnieniu zależności funkcyjnych wynikających z już określonych, reguły pozwalające stworzyć domknięcie zbioru zależności funkcyjnych. Są to np. reguły Armstronga (zwrotność, rozszerzenie, przechodniość).

22. Czym są nadmierne krotki (błędne) i jak przeciwdziałać ich powstawaniu?

Są to krotki powstające podczas dekompozycji stratnej, które nie występowały w oryginalnej relacji, zawierają fałszywe informacje i uniemożliwiają odtworzenie oryginalnej tabeli. Aby nie powstawały nie należy przeprowadzać dekompozycji stratnych.

Tura III

1. Dlaczego model ER nie może być stosowany do modelowania hurtowni danych?

Ponieważ najważniejszą cechą hurtowni danych jest ich szybkość działania, a model relacyjny ustępuje szybkością innym modelom.

2. Opisz strukturę organizacji hurtowni danych.

Składa się ze źródeł danych, które wysyłają dane do data loadera, który wczytuje dane do hurtowni w odpowiedni sposób (zmienia formaty danych, dostosowuje schematy, deduplikuje dane). Następnie zajmuje się nimi SZBD hurtowni, który udostępnia je narzędziom do zapytań i analizy danych. Same dane przechowywane są często w modelu innym niż relacyjny dla większej wydajności.

3. Co to jest free list?

Jest to lista wskaźników do wolnych obszarów pamięci w miejscach, gdzie przechowuje się rekordy o stałej długości. Powstają one poprzez usuwanie rekordów, a nowo wstawiane rekordy mogą szybko zostać wstawione na wolne miejsce dzięki wskaźnikowi z listy.

4. Czym jest system mediacyjny?

System mediacyjny przekształca dane z różnych źródeł w różnych formatach na jednolity schemat i format, co pozwala np. przechowywać je w jednej bazie danych i efektywnie wykorzystywać razem.

5. Podaj definicję 2NF.

Relacja jest w 2NF, jeśli jest w 1NF (wszystkie atrybuty są atomowe) oraz wszystkie niekluczowe atrybuty zależą od całych kluczy kandydujących (żaden nie zależy od tylko części jakiegoś klucza).

6. Podaj definicję BCNF. Czym różni się od 3NF? Dlaczego jest uważana za silniejszą?

Relacja jest w BCNF, jeśli dla wszystkich zależności funkcyjnych $A \rightarrow B$ zachodzi: A jest superkluczem lub $A \rightarrow B$ jest trywialne (B zawiera się w A).

Różni się od 3NF tym, że tam jest jeszcze “lub każdy atrybut z B jest częścią klucza kandydującego”, dlatego BCNF nakłada ostrzejsze ograniczenie i jest uważana za silniejszą.

Można powiedzieć, że w BCNF “dane zależą od klucza i tylko od klucza”, a w 3NF “niekluczowe dane zależą od klucza i tylko od klucza”.

7. Jakie są możliwe ograniczenia przy generalizacji/specjalizacji?

- 1) Które encje mogą być członkami danego zbioru encji niższego poziomu (jawny warunek/predykat).
- 2) Czy encje mogą należeć do więcej niż jednego zbioru encji niższego poziomu przy jednej generalizacji/specjalizacji (czy zbiory encji niższego poziomu muszą być rozłączne).
- 3) Czy zbiór pojęć ze zbioru encji wyższego poziomu musi należeć do przynajmniej 1 zbioru encji ze zbiorów encji niższego poziomu (czy zbiór encji wyższego poziomu musi być sumą zbiorów encji niższego poziomu). (ograniczenie kompletności)

8. Co to jest rozszerzenie tworzenia tabeli?

Jest to możliwość rozszerzenia polecenia CREATE TABLE o opcje utworzenia nowej tabeli jako identycznej do już istniejącej lub za pomocą wyrażenia SELECT (wtedy można też od razu wypełnić nową tabelę danymi z tego polecenia).

9. Wymień i opisz architektury systemów.

- Klient-serwer - dzieli zadania pomiędzy komputer klienta (żądający wykonania określonych zadań) a serwer (przechowujący informacje i dysponujący większą mocą obliczeniową),
- równoległe - wykorzystują współbieżność do przyspieszenia transakcji poprzez równoległe wykorzystanie wielu procesorów i dysków,

- rozproszone - rozpraszają swoje dane na wiele komputerów, gdzie określone dane są tworzone lub najbardziej potrzebne, równocześnie umożliwiając dostęp do nich z innych miejsc.

10. Jakie są różnice między dziedziną a własnym typem?

- Dziedziny mogą mieć ograniczenia takie jak not null czy default, a własne typy nie,
- typów używa się nie tylko do specyfikowania typu atrybutu, ale także w proceduralnych rozszerzeniach SQLa, gdzie może nie być możliwości wymuszenia ograniczeń,
- wartości jednej dziedziny można przypisać do wartości drugiej, o ile zgadzają się ich typy podstawowe.

11. Podaj definicję uogólnionej projekcji w algebrze relacji.

Uogólniona projekcja rozszerza projekcję pozwalając na użycie operatorów matematycznych w liście projekcji, np. w relacji z roczną wypłatą można uogólnioną projekcją dostać miesięczną wypłatę, bo można zrobić $\text{salary}/12$.

12. Jakies wyrażenie z selekcją i złączeniem naturalnym i je zinterpretować.

Złączenie naturalne działa jak iloczyn kartezjański z selekcją, w której wspólne atrybuty obu relacji mają mieć takie same wartości. Kolejne selekcje wystarczy dodać przed nawias.

13. Zapytanie z WITH i drugie zapytanie wykorzystujące ten WITH, zinterpretować wynik .

WITH pozwala zdefiniować wyrażenie SQL pod nową nazwą, używa się ich w innych kwerendach do uproszczenia zapisu. Jest to jak zmienna będąca wynikiem podzapytania.

14. Czym jest słownik danych i jaka jest jego organizacja?

Jest to specjalna tabela zawierająca metadane zwracane przez DDL zawierające informacje o danych z bazy danych i o jej strukturze (np. jej schemat), korzysta z niej głównie system zarządzania bazą danych. Słownik danych dla całej bazy

danych realizuje się jako miniaturową bazę danych wewnątrz właściwej bazy, zbiór tabel (często nieznormalizowanych dla wydajności).

15. Jakie zależności funkcyjne sprawiają najwięcej problemów?

Takie, w których lewa strona zależności jest podzbiorem klucza głównego oraz takie, w których lewa strona jest atrybutem niekluczowym.

16. Jak przechowywane są rekordy zmiennej długości?

Dla takich rekordów (z przynajmniej częścią atrybutów o zmiennej długości) najpierw przechowuje się pary (*offset,length*), gdzie *offset* to miejsce w pliku, gdzie zaczyna się atrybut i *length* to długość atrybutu w bajtach, potem alokuje się pamięć dla atrybutów o stałej długości, a następnie null bitmapę reprezentującą wartości null (jeśli bit jest 1, to odpowiednia wartość jest null) i na koniec atrybuty o zmiennej długości.

17. Dlaczego do klucza obcego potrzeba uprawnienia references?

Ponieważ uprawnienie references upoważnia do tworzenia relacji, które odnoszą się do danego klucza. Należy pamiętać, że jak utworzymy zależność do innego klucza, to nie będziemy mogli usunąć danego rekordu bez uprzedniego usunięcia odwołującego się rekordu

18. Jakie są typy atrybutów encji?

- Proste/złożone - atomowe lub składające się z wielu atomowych (np. adres),
- Jedno-/wielowartościowe - mające jedną wartość lub wiele wartości naraz (np. osoba i języki, którymi mówi),
- pochodne - wyliczane z innych atrybutów, nie przechowywane.

19. Do czego wykorzystywane jest zawężenie dziedziny?

Do ograniczenia dozwolonych wartości do pewnego podzbioru przez wyrażenie logiczne określające przedział lub za pomocą wyliczeniowej listy wartości.

20. Czym są blob i clob?

Blob to “binary large object”, duży plik zer i jedynek nieinterpretowany przez SZBD, tylko przez zewnętrzne programy z niego korzystające.

Clob to “character large object”, duży plik będący zbiorem znaków.

SZBD nie przekazuje obiektów blob i clob bezpośrednio, tylko wskaźniki do nich.

Tura IV

1. Podaj definicję zależności funkcyjnej. Do czego się ją stosuje?

Zależność funkcyjna to constraint pomiędzy dwoma zbiorami atrybutów, który wynika ze znaczenia i wzajemnych powiązań między atrybutami. Stosuje się ją do reprezentacji wymagania, aby wartość jednego zbioru atrybutów określała wartość innego zbioru atrybutów.

2. Podaj różnicę między algebrą relacji a językiem SQL.

SQL implementuje algebrę relacji wielozbiorów, tzn. pozwala na powtórzenia w wynikach, podczas gdy czysta algebra relacji eliminuje duplikaty z wyniku. Algebra relacji nie definiuje ponadto podzapytań, które występują w SQLu.

3. Jakie są podstawowe pojęcia w modelu ER? Wymień je i krótko opisz.

- Encja - reprezentacja rzeczywistego obiektu lub wydarzenia o określonych atrybutach (własnościach opisowych),
- zbiór encji - zbiór encji tego samego typu, o takich samych atrybutach,
- związek - połączenie między dwoma lub więcej encjami,
- zbiór związków - zbiór podobnych związków pomiędzy encjami, matematyczna relacja pomiędzy nimi, może mieć atrybuty.

4. Opisz cechy indeksu klastrowego.

- Zmienia fizyczną kolejność ułożenia rekordów,
- maksymalnie 1 na tabelę,
- maksymalnie na 1 kolumnę.

5. Co to są anomalie modyfikacji? Opisz i podaj przykład.

Są to takie przypadki, kiedy aktualizacja danych jednej krotki pociąga za sobą konieczność modyfikacji wartości atrybutów innych krotek.

Przykład: przechowujemy jako atrybut wartość, z której są obliczane inne wartości. Jeśli jego wartość się zmieni, to trzeba zaktualizować wartość wszystkich atrybutów z niego obliczanych.

6. Co to znaczy, że zależność jest trywialna?

Że zachodzi zawsze, nie może nie zachodzić, jest postaci $A \rightarrow B$, gdzie B zawiera się w A.

7. Wymień cechy związków w modelu ER.

- Są matematyczną relacją pomiędzy 2 lub więcej encjami,
- mogą mieć atrybuty,
- stopień - liczba encji biorących w nim udział,
- krotność - pomiędzy iloma encjami zachodzi związek, np. jeden-do-jeden, wiele-do-wiele,
- istnienie - czy jest częściowy, czy całkowity.

8. Jakie cechy mają widoki modyfikowalne?

- Klauzula FROM ma tylko jedną relację (widok dotyczy tylko 1 tabeli),
- klauzula SELECT jest prosta, tzn. zawiera tylko nazwy atrybutów i nie zawiera agregatów, wartości wyliczanych ani specyfikacji distinct,
- atrybuty niewyspecyfikowane w SELECT mogą być null'em lub mieć wartości domyślne,
- brak użycia GROUP BY i HAVING.

9. Podaj reguły dekompozycji do BCNF.

Najpierw musi być sprowadzona do 3NF (dziedziny wszystkich atrybutów atomowe, wszystkie atrybuty niekluczowe zależne tylko od kluczy kandydujących), a potem dla każdej nietrywialnej zależności funkcyjnej $A \rightarrow B$ zdekomponować relację R do relacji $(A \cup B)$ oraz $(R - (A \cup B))$.

10. Co to jest transakcja? Podaj jej cechy.

Transakcja to zbiór operacji, które wykonują jedno zadanie (jednostka logiczna działań na bazie). Prawidłowa transakcja musi przestrzegać zasad ACID:

- 1) Atomicity (atomowość, niepodzielność) - transakcja to atomowa całość, “wszystko albo nic” - następuje cała albo w ogóle nie jest realizowana.
- 2) Consistency (spójność) - transakcja nie może naruszać zasad integralności systemu, po jej wykonaniu baza danych ma spełniać wszystkie swoje warunki integralności (przekształca stan spójny na stan spójny).
- 3) Isolation (izolacja) - jeśli dwie transakcje są wykonywane współbieżnie, to nie widzą wprowadzanych przez siebie zmian (działają w izolacji).
- 4) Durability (trwałość) - po wykonaniu transakcji system musi prawidłowo uruchamiać się i udostępniać spójne, nienaruszone, aktualne dane zapisane po zatwierdzonych transakcjach, także w wypadku krytycznego błędu (np. nagłej utraty zasilania).

Pytania różne (autorskie, 2016/2017 i wcześniejsze etc.)

1. Co to jest SZBD?

System Zarządzania Bazami Danych (SZBD) to zbiór programów pozwalający użytkownikom na tworzenie i utrzymanie baz danych. Pozwala na kontrolę redundancji, normalizację danych, ułatwienie zarządzania bezpieczeństwem i wielodostęp do danych.

2. W jaki sposób w SQLu tworzy się klucz kandydujący? Czy może on przyjmować wartość null?

Poprzez użycie unique na atrybucie lub krotce atrybutów (A_1, A_2, \dots, A_n). Tak jak klucz główny nie może zawierać wartości null (trzeba na to constraintów).

3. Czy klucz obcy musi odwoływać się do klucza głównego innej tabeli?

Nie, może odwoływać się do klucza kandydującego z innej tabeli (krotki oznaczonej jako unique).

4. Czy można stworzyć kolumnę unique default?

Można, jednak przy wstawieniu już drugi raz rekordu korzystając z ustawiania wartości domyślnej zostanie wyrzucony błąd.

5. Co należy zrobić, kiedy chce się wstawić do tabeli dane nieprzestrzegające nałożonego na jej atrybuty constrainta?

Wstawić dane używając WITH NOCHECK, zmienić dane tak, aby przestrzegały constrainta (np. odpowiednią procedurą), a potem uaktywnić constraint.

6. Jakie constrainty można blokować na istniejących tabelach?

Check i foreign key, inne trzeba usunąć i dodać z powrotem.

7. Czy widoki to część modelu logicznego bazy danych?

Nie, są "wirtualnymi relacjami" nieistniejącymi w modelu logicznym.

8. Czym są widoki zmaterializowane? Jaki jest ich główny problem i jak można sobie z nim radzić?

Są to fizyczne relacje (tabele) zawierające krotki będące wynikami zapytań definiujących widok. Ich główny problem to potrzeba utrzymywania widoku aktualizacji danych, służą do tego np. programy, triggerzy, utrzymywanie przyrostowe (całościowe/częściowe, natychmiastowe/opóźnione).

9. Jakie są zalety i wady wykorzystywania widoków zmaterializowanych w porównaniu ze zwykłymi widokami?

- Widoki zmaterializowane zapewniają szybszy dostęp do zapytań wykorzystujących agregaty i obliczenia na dużych zbiorach danych lub złożone złączenia,
- kiedy widok miałby być wykonywany często, lepiej stworzyć widok zmaterializowany i wykonać zapytanie raz,
- zwykłe widoki nie wymagają kosztów wydajnościowych na utrzymywanie,
- materializacja wymaga miejsca na dysku.

10. Czy wstawienie krotki do widoku zawsze się powiedzie?

Nie, gdyż niektóre operacje wstawiania mogą być niejednoznaczne (nie będzie wiadomo, do których tabel tworzących widok dodać nowe informacje) lub wstawić błędne informacje, dlatego wiele SZBD nie pozwala na to lub pozwala na to tylko bardzo prostym widokom (patrz pytanie 17 z pytań z eksperymentu dydaktycznego).

11. Czym jest trigger?

Trigger to procedura wykonywana automatycznie jako reakcja na określone zdarzenia w bazie danych, mogą np. ograniczać dostęp do danych, nadzorować modyfikacje lub rejestrować zmiany.

12. Jak zbudowany jest plik indeksowy?

Jest przechowywany jako struktura B⁺-drzewa, wydajniejsza od wcześniej używanych plików sekwencyjnych (tabel z parami (search-key,pointer)). Struktura ta sama się reorganizuje, nie wymaga całościowych reorganizacji i jest wydajna kosztem tylko niewielkiego narzutu pamięciowego.

13. Czy w SQLu użytkownik może tworzyć także złożone własne typy danych?

Tak, tzw. “structured data types”, mogą one zawierać zagnieżdżone rekordy, tabele czy wielozbiory.

14. Jakie korzyści daje zdefiniowanie własnego typu danych, kiedy jest on zdefiniowany za pomocą prostych typów wbudowanych, np.:

CREATE TYPE Dollar AS numeric(12,2) final?

Zgodność typów jest sprawdzana dla nowego typu, więc próbując przypisać wartość numeric(12,2) do atrybutu typu Dollar otrzyma się błąd. Daje to więc większą kontrolę nad danymi.

15. Wymień i opisz formy uprawnień do części bazy danych.

- Read - odczyt danych,
- Insert - wstawianie nowych danych,
- Update - modyfikacja istniejących danych,
- Delete - usuwanie danych.

16. Wymień i opisz formy uprawnień do schematu bazy danych.

- Index - tworzenie i usuwanie indeksów,
- Resources - tworzenie nowych relacji (tabel),
- Alteration - dodawania i usuwanie atrybutów w relacji,
- Drop - usuwanie relacji.

17. Czy twórca widoku, funkcji lub procedury dostaje do niej wszystkie uprawnienia?

Nie, dostaje tylko takie, których nie wymagają dodatkowych ponad te, które już ma.

18. Podaj przykłady poleceń w języku SQL nadających i odbierających uprawnienia.

GRANT <privilege list> ON <relation/view name> TO <user/role list>

REVOKE <privilege list> ON <relation/view name> TO <user/role list>

Privilege list to jedno z: select, insert, update, delete, all privileges.

User list może zawierać po prostu public, dotyczy wtedy wszystkich użytkowników (obecnych i przyszłych).

19. Co łatwiej zmienić w istniejącej bazie danych i dlaczego - schemat fizyczny czy logiczny?

Łatwiej jest zmienić schemat fizyczny, bo dzięki abstrakcji (niezależności danych od programu) można łatwo zmieniać fizyczne reprezentacje bazy danych bez wpływania na schemat logiczny, a zmiana schematu logicznego pociąga za sobą konieczność przepisania istniejących zapytań.

20. Po co w ogóle korzysta się z modelu ER zamiast np. od razu modelu logicznego?

Ponieważ pozwala łatwo odwzorować schemat organizacji oraz znaczenia rzeczywistych obiektów i interakcji na model konceptualny, dzięki temu potem można szybko i łatwo stworzyć pozbawiony błędów model logiczny.

21. Zdefiniuj bazę danych za pomocą pojęć modelu ER.

Baza danych to kolekcja zbiorów encji, z których każdy posiada dowolną ilość encji odpowiedniego typu.

22. Jaki jest minimalny możliwy stopień związku zbiorów encji? Czy jest on typowy?

1, bo może być rekursywny (a jeśli nie jest wiele-do-wiele, to nie wymaga tabeli łącznikowej). Nie jest typowy, bo większość związków ma stopień 2 (związki binarne).

23. Czym różni się atrybut złożony od wielowartościowego i w jaki sposób się je przechowuje?

Atrybut złożony składa się z innych atrybutów, np. adres składa się z 3 atrybutów (miasto, ulica, numer domu).

Atrybut wielowartościowy natomiast to pojedynczy atrybut przechowujący wiele wartości - wydaje się, że wymaga przechowywania wielu różnych wartości naraz, np. chcąc wyrazić języki, którymi mówi pracownik można by stworzyć wielowartościowy atrybut SpokenLanguages zawierający wszystkie języki, którymi mówi (każdy język to wartość).

Atrybuty złożone rozbija się na atrybuty atomowe, tworząc z nich nową tabelę, a zamiast atrybutu wielowartościowego tworzy się tabelę zawierającą jego kolejne wartości jako rekordy.

24. W jaki sposób rozróżnia się encje w zbiorach encji?

Za pomocą kluczy złożonych z ich atrybutów. Taki zbiór (może być ich wiele) to superklucz, minimalny zbiór atrybutów tworzący superklucz to klucz kandydujący, a spośród nich wybiera się jeden jako klucz główny, który dalej wykorzystuje się do rozróżniania.

25. W jaki sposób rozróżnia się związki w zbiorach związków encji?

Za pomocą kluczy, gdzie superklucze tworzy się za pomocą kombinacji kluczy głównych zbiorów encji biorących udział w związku, a następnie wybiera się minimalne superklucze (klucze kandydujące) i spośród nich klucz główny. Dla związku wiele-do-wiele superklucz tworzy zbiór wszystkich kluczy głównych zbiorów biorących udział w związku, dla związku wiele-do-jeden klucz główny zbioru po stronie jeden.

26. Czy wszystkie związki n-arne można przekształcić na binarne z zachowaniem wszystkich ograniczeń?

Nie, bo choć wszystkie związki n-arne można przekształcić na binarne, to niektóre ograniczenia nie dają się przetłumaczyć przy takiej zmianie.

27. Jak w modelu ER realizuje się kompozycję?

Jako związek wiele-do-jednego z ograniczeniem integralności liczności.

28. Jeśli tabela ma udział obowiązkowy w związku z inną tabelą (powiązaną), to czy zawsze można wstawić rekordy do tej tabeli powiązanej?

Nie, w takiej sytuacji do tabeli powiązanej można wstawiać rekordy dopiero wtedy, kiedy w tabeli z udziałem obowiązkowym jest przynajmniej 1 rekord.

29. Jeśli mamy tabelę z 3 rekordami powiązaną z inną tabelą związkiem o stopniu uczestnictwa (4,11), to czy do tej drugiej tabeli można wstawić rekordy? Dlaczego?

Nie, bo stopień uczestnictwa (4,11) oznacza, że rekord w drugiej tabeli jest powiązany z minimum 4 i maksymalnie 11 rekordami z pierwszej tabeli, więc 3 jest niewystarczające.

30. Czym różnią się języki proceduralne od deklaratywnych?

W językach proceduralnych (np. algebra relacji) użytkownik definiuje, jak krok po kroku wykonać zadanie, a w językach deklaratywnych (np. logika predykatów) deklaruje tylko jaki rezultat chce osiągnąć, a nie jak go osiągnąć.

31. W jaki sposób w algebrze relacji zrealizować modyfikację?

Poprzez użycie operacji przypisania na stałej relacji.

32. Wymień 5 rodzajów złączeń (join'ów) z algebry relacji.

Natural join, left outer join, right outer join, full outer join, theta join.

33. Czy w algebrze relacji przy operacji agregacji można grupować po pustej liście atrybutów?

Tak, można.

34. Czy w algebrze relacji przy operacji agregacji można w jednym użyciu operatora używać wielu funkcji agregujących na wielu różnych atrybutach?

Tak, można.

35. W jaki sposób (za pomocą jakich operatorów) w algebrze relacji można odwzorować modyfikacje bazy danych (wstawianie, usuwanie, modyfikacja)?

Wstawianie: $r \leftarrow r \cup E$

Modyfikacja: $r \leftarrow \Pi_{F_1, F_2, F_3 \dots}(r)$

Usuwanie: $r \leftarrow r - E$

Używa się modyfikatora przypisania w połączeniu z modyfikatorem sumy, projekcji uogólnionej lub różnicy.

36. W algebrze wielozbiorów jeśli na wejściu w zbiorze pierwszym było m duplikatów, a w drugim n duplikatów, to ile duplikatów będzie w ich iloczynie kartezjańskim?

Będzie ich $m \cdot n$.

37. Jakie dodatkowe operatory wprowadza się do algebry relacji dla upraszczania zapytań (nie dają one nowych możliwości)?

Przecięcie zbiorów, złączenie naturalne, przypisanie i złączenie zewnętrzne.

38. Jakie są konsekwencje redundancji danych?

Marnowanie miejsca oraz występowanie anomalii (wstawiania, modyfikacji i usuwania).

39. Jakiego rodzaju złączenia powinien dopuszczać schemat bazy danych, aby nie pozwalać na tworzenie błędnych krotek?

Tylko złączenia w oparciu o porównywalne atrybuty (klucze główne i obce).

40. Do czego służy teoria zależności funkcyjnych?

Do formalnej analizy schematów dla wykrywania w sposób formalny problemów, np. redundancji danych czy anomalii. Jest ona oparta o klucze, zależności funkcyjne i wielowartościowe.

41. Czym jest wyznacznik zależności funkcyjnej i co on determinuje?

W zależności funkcyjnej $A \rightarrow B$ wyznacznikiem jest A (lewa strona zależności), determinuje on wartości B (prawa strona zależności, funkcyjnie zależna).

42. Czy zależność funkcyjna może zachodzić tylko dla wybranych instancji relacji?

Nie, musi obowiązywać dla wszystkich instancji relacji, nazywa się je wtedy legalnymi.

43. Czym jest domknięcie zbioru zależności funkcyjnych i jakie 4 zasady pomagają praktycznie je obliczyć?

Jest to zbiór wszystkich zależności funkcyjnych, które wynikają ze zbioru zależności funkcyjnych (można je z niego obliczyć).

Zasady (gdzie $AB = A \cup B$):

1. Samookreślenie: $A \rightarrow A$
2. Rozkład: jeżeli $A \rightarrow BC$, to $A \rightarrow B$ i $A \rightarrow C$
3. Suma: jeżeli $A \rightarrow B$ i $A \rightarrow C$, to $A \rightarrow BC$
4. Złożenie: jeżeli $A \rightarrow B$ i $C \rightarrow D$, to $AC \rightarrow BD$

44. Zdefiniuj superklucz i klucz kandydujący za pomocą pojęcia zależności funkcyjnej.

Zbiór atrybutów A należący do relacji R jest jej superkluczem wtedy i tylko wtedy, gdy zachodzi zależność funkcyjna $A \rightarrow R$.

Zbiór atrybutów A należący do relacji R jest jej kluczem kandydującym wtedy i tylko wtedy, gdy zachodzi zależność funkcyjna $A \rightarrow R$ oraz dla żadnego podzbioru B zbioru A nie zachodzi $B \rightarrow R$.

45. Czym jest pełna zależność funkcyjna?

Jest to taka zależność $A \rightarrow B$, że usunięcie dowolnego atrybutu z A sprawia, że zależność już nie zachodzi.

46. Czym jest przechodnia zależność funkcyjna?

Jest to taka zależność $A \rightarrow B$, że w relacji istnieje zbiór C , który nie jest kluczem kandydującym ani podzbiorem klucza oraz zachodzi $A \rightarrow C$ i $C \rightarrow B$.

47. Czy w procesie normalizacji zawsze trzeba najpierw usunąć częściowe zależności funkcyjne (redukując je do pełnych), a dopiero potem wyeliminować zależności przechodnie?

Nie trzeba, takie działanie to pozostałość historyczna (bo 3NF definiowano za pomocą 2NF).

48. Zdefiniuj zależność wielowartościową.

Reprezentuje ona taki związek zbiorów atrybutów A, B, C, że dla każdej wartości z A istnieją dwa całkowicie niezależne od siebie zbiory wartości B i C (A wyznacza funkcyjnie zarówno B, jak i C), oznacza się to jako $A \twoheadrightarrow B$, $A \twoheadrightarrow C$.

49. Zdefiniuj 4NF.

Relacja R jest w 4NF, jeśli jest ona w BCNF oraz dla każdej zależności wielowartościowej $A \twoheadrightarrow B$ jest ona trywialna (B zawiera się w A lub suma A i B daje całe R) lub A jest superkluczem.

50. Czym jest zależność złączeniowa?

Jest to sytuacja, kiedy schemat jest w 4NF oraz istnieje jego bezstratny rozkład na 3 lub więcej relacji. Dotyczy sytuacji, kiedy w relacji występują co najmniej dwie zależności wielowartościowe.

51. Zdefiniuj 5NF.

Relacja R jest w 5NF, jeśli jest ona w 4NF oraz nie występują w niej zależności złączeniowe, tzn. nie da się jej bezstratnie rozbić na mniejsze relacje, a każdy schemat składa się z atrybutów tworzących superklucz.

52. Kiedy lepiej używać atrybutów o zmiennej długości, a kiedy o stałej?

Atrybuty o zmiennej długości pozwalają zaoszczędzić miejsce, kiedy dany atrybut tylko czasami przyjmuje bardzo długie wartości - analogiczny atrybut stały także dla krótszych wartości alokowałby miejsce takie, jakiego wymaga rekord o największej długości.

Atrybutów o stałej długości lepiej jest używać zawsze wtedy, kiedy nie jest wymagane użycie atrybutu o zmiennej długości, bo stała długość wymaga mniej mocy obliczeniowej (jest szybciej).

53. Czym jest wielodostęp? Jakie może powodować problemy?

Jest to wykonywanie równolegle wielu transakcji. Może powodować problemy związane z taką współbieżnością, np. wzajemne przestanianie modyfikacji, wykorzystywanie wzajemnych częściowych (a nie finalnych) wyników, odczytywanie danych w trakcie ich modyfikacji, odczytywanie różnych wartości tego samego atrybutu w różnym czasie (przez jego modyfikację przez inną transakcję), pojawianie się krotek fantomowych (nowych krotek wstawionych przez inną transakcję).

54. W jaki sposób przeciwdziałać problemom z wielodostępem?

Aby w pełni zachować zasady ACID (w szczególności izolację), należy je szeregować poprzez ułożenie odpowiedniego harmonogramu sekwencyjnego, realizuje się to poprzez ułożenie grafu pierwszeństwa. Wykorzystuje się także blokady odpowiednich danych ze znacznikami czasowymi lub testy walidacyjne (sprawdzanie integralności pod koniec transakcji).