

01. Kdy se začal formovat koncept relačních databází (Vznik relačního modelu, první definice SQL)?

1969 (relační model), 1970 (SQL)

(zdroj: [Wikipedia](#)), 4. slide z 1. prednasky ("1969 – Codd - Relační model")

02. Kdy přibližně vznikly první komerční relační databázové servery?

1980

(zdroj: [Wikipedia](#)), 4. slide z 1. prednasky ("1979 – Oracle 2, basic SQL, no transaction")

03. K čemu slouží Data management v organizaci?

- a) K definici organizační struktury správců dat
 - b) Ke kategorizaci dat a předávání dat mezi jednotlivými složkami organizace**
 - c) K zajištění bezpečnosti dat
 - d) Ke starosti o data jako o jiné druhy majetku organizace
-

04.

Jaký je rozdíl mezi daty a informacemi?

- Data jsou surová „data“, která nám nedávají vůbec žádné další informace. Informace už nám pomáhají pochopit jednotlivé vztahy mezi daty – Kdo? Co? Kde? Kdy?
- Data + (definition, format, timeframe, relevance) = Information

Jaký je rozdíl mezi a informacemi a znalostmi?

- Oproti informacím nám pomáhají pochopit i jak?. Porozumění vzorců
- Information + (patterns & trends, relationships, assumptions) = Knowledge

(druhé odpovědi (anglicky), zdroj: MI-DSP, prednaska 1, slide 14)

05. Uveďte alespoň dva způsoby, jak je možné dělit data v organizaci.

- Organizační struktury (oddělení datové kvality, oddělení bezpečnosti, Data owner)
- Kultura organizace
- Plán vývoje a údržby (IT architektura, Datová architektura)

(zdroj: MI-DSP, prednaska 1, slide 13)

06. Co to je datová kvalita? Jak se pozná, že jsou data kvalitní?

- Vlastnost dat, která není daná jejich strukturou nebo uložením.
- Dostupnost, úplnost, bezpečnost, srozumitelnost atd.

(zdroj: MI-DSP, prednaska 1, slide 16)

07. Uveďte čtyři způsoby, jak je možné přistoupit k zlepšení kvality dat.

1. Ignorovat znečištění dat – přeneseme problém na uživatele.
2. Jednorázově vyčistíme (přefiltrování vody).
3. Použití pouze aktuálních dat (filtrace jen aktuálních dat).
4. Předejít budoucím chybám (najít původce znečištění).

(zdroj: MI-DSP, prednaska 1, slide 21)


08. Co to jsou metadata?

- Metadata jsou data popisující data.
- Strukturované informace, které nám umožňují najít informace o datech, spravovat je, kontrolovat je, porozumět jim.

(zdroj: MI-DSP, prednaska 1, slide 24)

09. Uveďte, jaké typy metadat znáte.

- Popisná.
- Administrativní.
- Strukturální.
- Technická.
- Aplikační.

Mnemotechnická pomůcka: PASTA :) 

(zdroj: MI-DSP, prednaska 1, slides 27, 28).

10. Uved'te čtyři základní činnosti, které se provádí s metadaty v organizaci.

- Shromažďování.
- Integrace a ukládání.
- Analýza.
- Prezentace.

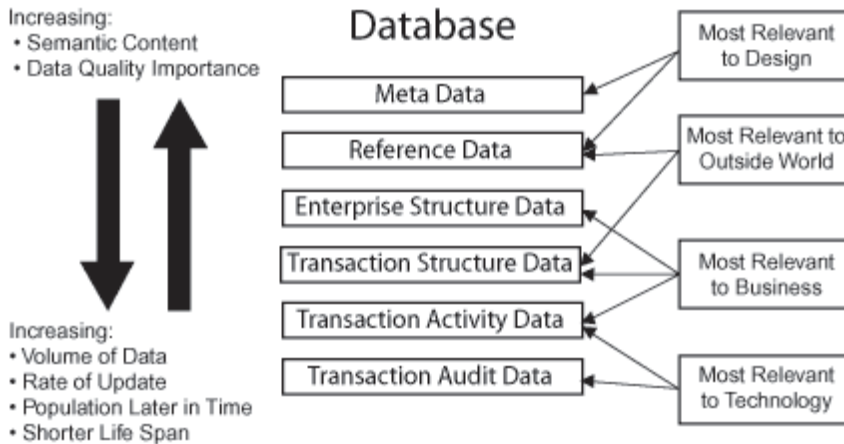
(zdroj: MI-DSP, prednaska 1, slide 29)

11. Popište dělení dat do vrstev podle Malcolma Chisholma.

6 vrstev databáze

- směrem nahoru roste sémantický význam a důraz na kvalitu dat
- směrem dolů roste objem dat, počet aktualizací dat a klesá životnost

1. Metadata
2. Reference Data
3. Enterprise Structure Data
4. Transaction Structure Data
5. Transaction Activity Data
6. Transaction Audit Data



Pokud vám to nedává moc smysl, tak tu:

<http://www.information-management.com/issues/20010901/3985-1.html>

(zdroj: MI-DSP, prednaska 1, slide 12,

<http://www.information-management.com/issues/20010901/3985-1.html>)

12. K čemu slouží metadata?

- a) K porozumění mezi uživateli dat a informací
 - b) K plánování zálohování a obnovy dat
 - c) K ohodnocení ceny uložených dat a celkového TCO
 - d) K zjednodušení dopadových analýz
-

13. Co patří mezi základní cíle RDMS?

- a) Uložení datových struktur a dat
 - c) Zabezpečení uložených dat
 - d) Vytvoření uživatelského prostředí pro práci s daty
-

14. Seřadte prostředky HW podle důležitosti pro výkon DBMS? Odůvodněte svoje ohodnocení.

- a) RAM
- b) Diskové prostory
- d) Sběrnice
- c) Procesory
- e) Síťový subsystém

DBMS používají jako uložení především disky + indexy a často dotazovaná data jsou v ramkách, takže rychle disky (např. SSD) a to kvůli velké ramce pro uložení indexu, protože na vyhledání dat, sběrnici jako lepidlo a síť pro spolehlivý a rychlý přenos výsledku

15. Které paralelní architektury DBMS znáte?

- Shared memory
 - Shared disk
 - Shared nothing
 - NUMA
-

16. Popište hlavní rysy Shared Memory architektury.

- procesory a disk mají přístup do společné paměti
 - vysoce efektivní komunikace mezi procesory – data jsou přístupná všem procesorům
-

17. Popište hlavní rysy Shared nothing architektury RDMS.

- každý procesor má vlastní paměť i data – data jsou zpracována pouze příslušným procesorem (nepředávají se mezi nody)
 - nody spolu komunikují přes síť, předávají si požadavky a výsledky
-

18. Které databázové servery používají architekturu Shared Memory?

- a) Oracle 11gR2
 - b) Teradata
 - c) Microsoft SQL Server
 - d) MySQL
 - e) Tandem
-

19. Které databázové servery používají architekturu Shared nothing?

- a) Oracle 11gR2
 - b) Teradata**
 - c) Microsoft SQL Server
 - d) MySQL
 - e) Tandem**
-

20. Architekturu klient-server nejlépe vystihuje tvrzení:

- a) Jeden server vyřizuje požadavky pro různé druhy klientů**
 - b) Pro server je dedikovaný hardware
 - c) Komunikace mezi klientem a serverem probíhá v SQL
 - d) Klienti mohou být málo výkonné stanice
-

21. Co je v současnosti kritický limit pro výkonnost RDMS?

- a) Rychlost sítí
 - b) Rychlost interních sběrnic
 - c) Zastaralost operačních systémů
 - d) Rychlost disků**
 - e) Velikost a cena operační paměti
 - f) Rychlost procesorů
 - g) Malá podpora paralelního zpracování procesorů
-

22. Popište životní cyklus uživatelského požadavku v prostředí klient/server.

- Navázání spojení s klientem
 - Porozumění požadavku
 - Optimalizace a vytvoření výpočtu
 - Vlastní výpočet
 - Předání výsledků
-

23. Jaký je rozdíl mezi RULE based a COST based optimalizátorem SQL dotazu?

- rule-based vyhodnocuje jednotlivé přístupové cesty pomocí předem daného systému pravidel
 - cost-based optimizer hledá plán s nejmenšími „náklady“ (využívá zejména Oracle)
-

24. Co jsou hlavní úkoly parseru SQL dotazu v RDMS?

- a) Provést syntaktickou analýzu dotazu**
 - b) Určit kde jsou uložena potřebná data
 - c) Namapovat SQL dotaz na objekty v databázi
 - d) Předat informaci o stavu dotazu uživateli
 - e) Vybrat pořadí spojování tabulek v joinech
-

25. Co jsou hlavní úkoly optimalizátoru SQL dotazu v RDMS?

- a) Provést syntaktickou analýzu dotazu
 - b) Udržovat statistiky nutné pro optimalizaci
 - c) Vybrat přístupové metody k datům**
 - d) Vybrat pořadí spojování tabulek v joinech**
-

26. Jaké typy informací předává RDMS server klientovi?

- a) Požadovaná data**
 - b) Informace o formátu předávaných dat**
 - c) Chybové zprávy**
 - d) Informace o průběhu zpracování
 - e) Statistiky o požadovaných datech
-

27. Proč a jak se ověřuje konzistence datových struktur uložených dat?

- už z definice ACID plyne, že data uvnitř DB by měla být konzistentní -> být vždy smysluplná. Nekonzistentní stav DB by uváděl například záporný věk uživatele.
 - kontrola pomocí kontroly interních struktur, indexů
 - kontrola pomocí výpočtu checksum pro zapsané bloky
-

28. K čemu se RDMS používají paměť RAM?

- a) Jako datovou keš**
 - b) Pro správu interních struktur RDMS
 - c) Pro ukládání mezivýsledků výpočtů**
 - d) Pro komunikaci s klienty
 - e) Pro kešování kódu uložených procedur a triggerů
 - f) Pro kešování výsledků**
 - g) Pro komunikaci mezi jednotlivými procesy**
 - h) Pro ukládání transakčního logu
-

29. Popište LRU algoritmus pro správu datové keše.

LRU (Least recently used) vyhazuje z paměti stránku, která nebyla nejdelší dobu použita. Má-li být z paměti některá stránka vyhozena, vybere se ta, která nebyla použita nejdéle.

30. Jaká je vazba mezi procesy RDMS spracovávající požadavky klientů a procesy operačního systému, na kterém běží RDMS.

- a) Co klient to proces
 - b) Co klient to thread
 - c) Jeden proces RDMS v operačním systému obsluhuje více klientů
 - d) Jak u kterého RDMS**
-

31. Transakční log slouží k

- a) Zápisu historie požadavků klientů
- b) K obnově konzistentního stavu po výpadku serveru**
- c) Umožňuje při operaci rollback přejít k původnímu stavu**
- d) Zajištění atomicity transakcí

(alespoň orientační zdroj [zde](#))

32. Co to je operace checkpoint?

- a) Operace, kdy se zapíše transakční log na disk
 - b) Operace, kdy se zapíše obsah změněných stránek v keši na disk**
 - c) Operace, kdy se zkontroluje, že zápis databáze na disk je konzistentní
 - d) Ani jedna z uvedených operací
-

33. Popište Write-ahead log model:

- WAL zajišťuje atomicity a durability.
 - Před uložením bloku dat do databáze se musí nejdříve uložit všechny změny do logu a pak až se promítnou do DB.
 - Zajišťuje schopnost zotavení systému po výpadku.
-

34. Popište, jak probíhá recovery databáze po pádu serveru při použití Write-ahead modelu transakčního logu.

- podívá se do transakčního logu, který musí být uložen dříve, než se začnou provádět změny nad DB a podle toho se DB upraví, aby opět byla konzistentní.
 - se všemi necommitovanými transakcemi je proveden rollback
-

35. Uveďte čtyři kroky vedoucí k vytvoření relačního databázového schématu.

- Shromáždění business požadavků.
 - Konceptuální model.
 - Logický model.
 - Fyzický model.
-

36. Co je cílem sběru požadavků při vytváření datového modelu.

- pochopit business doménu (Data-flow diagram)
-

37. Co to je dataflow diagram a k čemu slouží.

- diagram, který poskytuje lepší pochopení procesů uvnitř firmy.
- Kdo data vytváří, kdo je zpracovává, kde jsou uložena, kdo je používá atd.

38. Co je cílem při vytvoření konceptuálního modelu.

- určit entity a jejich atributy
 - u atributů určit jejich hodnoty a vlastnosti
 - u entity určit jejich potenciální klíče a identifikovat vazby mezi entitami
-

39. Na jakém základě se definuje počet a granularitu entit v konceptuálním modelu?

- a) Na základě sebrané požadavků
 - b) Na základě standardních patternů a modelů
 - c) Na základě schopností a možností použitého RDMS
 - d) Libovolně, přesná definice vzniká až na úrovni logického datového modelu.**
-

40. Konceptuální model obsahuje:

- a) Relace**
 - b) Atributy**
 - c) Datové typy atributů
 - d) Kardinalitu relací**
 - e) Primární klíče
 - f) Normalizované entity
-

41. Popište rozdíl mezi potenciálním klíčem entity a primárním klíčem entity.

(IMHO) Potenciální klíč je jen klíč, který se potenciálně nabízí být klíčem primárním, takže třeba i umělý klíč.

Potenciální klíč je takový klíč, který by mohl být primárním klíčem v tabulce, kde ale už existuje jiný primární klíč (př. u studenta máte v tabulce atribut evidenční číslo a atribut rodné číslo, obě jsou unikátní, EČ je PK a RČ je potenciální klíč - mohl by být primárním)

42. Popište rozdíl mezi primárním klíčem a primárním indexem.

- **IMHO je to to samé, akorát česky to je klíč a anglicky index. V tabulce nelze mít dva záznamy se stejným klíčem. Primární index se vytvoří vždy tam, kde je v tabulce nastaven primární klíč. Kromě primárního indexu můžeme indexovat i další sloupce tabulky, například kvůli rychlejšímu přístupu k datům. - VIZ COMMENT**
-

43. Popište rozdíl mezi závislými a nezávislými entitami (Relace závislost).

- pokud relace A je závislá na B, musí být v relaci A uveden cizí klíč na relaci B.
 - jasně definujeme, že relace A je závislá na nějakém záznamu z relace B
-

44. Uveďte příklady grafických notací relací pro konceptuální model.

-----o--/CAR/ Nejvíce jedno auto

45. Co je výstupem konceptuálního modelování?

a) Entity-relation diagram

b) Datový model

c) Data-flow diagram

d) Seznam entit, atributů a relací společně s jejich podrobným popisem

46. Co je cílem při vytvoření logického datového modelu?

a) Vytvořit Entity-relation diagram

b) Vytvořit Data-flow diagram

c) Vytvořit platformně nezávislý logický datový model

d) Vytvořit model v RDMS

47. Pro vytváření logického datového modelu je třeba:

a) Převést model do třetí normální formy (normalizace)

b) Převést relace na cizí klíče (normalizace)

c) Definovat primární klíče

d) Rozhodnout o reprezentaci subtypů

e) Definovat vazby mezi entitami

f) Definovat datové typy atributů

g) Navrhnout vhodné indexy

48. Doplňte tabulku (jména sloupců a data) tak, aby nesplňovala první normální formu.

1NF: Všechny sloupce nelze dále dělit na části

<u>AUTHOR_ID</u>	PHONE #1
101	111222333, 444555666
102	777888999, null

49. Doplňte tabulku (jména sloupců a data) tak, aby splňovala první normální formu a nesplňovala druhou normální formu.

2NF: Každý neklíčový atribut je plně závislý na celém primárním klíči (ne jen na části klíče)

<u>AUTHOR_ID</u>	<u>BOOK_ID</u>	GENRE
101	201	scifi
102	201	scifi
103	202	roman
104	202	roman

50. Doplňte tabulku (jména sloupců a data) tak, aby splňovala druhou normální formu a nesplňovala třetí normální formu.

3NF: Nezávislý atribut nesmí tranzitivně záviset na neklíči (jinak řečeno, všechny neklíčové atributy jsou navzájem nezávislé)

<u>BOOK_ID</u>	TITLE	CATEGORY	CATEGORY_DESCRIPTION
201	trol	301	fantasy
202	yyy	301	fantasy
203	zzzz	302	novel
204	rrrr	302	novel

Všechny neklíče jsou závislé na BOOK_ID ale neklíč CATEGORY_DESCRIPTION je navíc závislý na neklíči CATEGORY

51. Uveďte alespoň tři kritéria, které je nutno brát v úvahu při výběru primárního klíče:

- musí mít vždy definovanou hodnotu (not null)
 - musí mít stálou hodnotu (během celého životního cyklu řádku)
 - musí být co možná nejmenší (co nejméně sloupců)
-

52. Popište (nakreslete) jak se převádí relace typu N:N z E-R diagramu na tabulky.

- Pomůžeme si třetí pomocnou tabulkou, která je s oběma původními tabulkami spojena vztahem 1:N. Celé to pak simuluje N:N



53. Mezi důvody normalizace schématu patří:

- a) Snížení hodnot null v datech
 - b) Zajištění business pravidel
 - c) Snížení redundance dat
 - d) RDMS neumí pracovat s nenormalizovaným modelem
 - e) Zjednodušení správy dat
 - f) Zjednodušení dalších úprav modelu
-

54. Popište, co to je funkční závislost mezi sloupci logického datového modelu.

- Jestliže platí pro kterékoliv dva řádky že pokud se shoduje hodnota ve sloupci A, potom už se shoduje hodnota i ve sloupci B, říkáme
 - A funkčně určuje B
 - B je funkčně závislé na A

55. Co je cílem vytvoření fyzického datového modelu.

- vytvořit fyzický model s ohledem na specifika aplikace a použitý typ databáze, použitý hardware (tabulky, datové typy ...)

56. Které činnosti je třeba vykonat při převodu logického datového modelu na fyzický

- a) Určit relace mezi tabulkami
- b) Určit datové typy atributů**
- c) Definovat primární klíče
- d) Definovat referenční integritu**
- e) Definovat primární indexy**
- f) Vytvořit procesní matici**
- g) Provést denormalizaci modelu**

57. Při stanovení jmenných konvencí fyzického datového modelu je třeba brát v úvahu:

- a) Omezení konkrétního RDMS – například case sensitive / case insensitive**
- b) Porozumění modelu – datové tabulky, číselníky, logy**
- c) Vazbu na konceptuální datový model
- d) Datové typy atributů
- e) Typy tabulek, indexů a dalších objektů v databázi

58. Jaká je vazba mezi referenční integritou a cizím klíčem v databázi.

- referenční integritu bereme v úvahu pouze při logickém datovém modelu. Ve fyzickém tuto skutečnost modelujeme už pomocí cizích klíčů. Jinak je to to samé.

59. Jaké typy denormalizace znáte:

- partitioning (horizontální / vertikální)
- uložení vypočtených hodnot
- spojení tabulek

60. Jaký je rozdíl mezi vertikální a horizontální denormalizací.

- horizontální: tabulka se rozdělí podle záznamů. Nejsou přístupné všechny záznamy
- vertikální: tabulka se rozdělí podle sloupců. Nejsou přístupné všechny informace o záznamech

61. Napište alespoň tři důvody, proč se přistupuje k denormalizaci datového modelu.

- ušetření nákladných dotazů (join) při spojování tabulek
 - Neustálé dotahování hodnot z číselníků
 - Omezení datových serverů (Sybase třicet tabulek v jednom joinu)

- Suptype/supertype vazba
 - vytvoření tabulky v případě, že často přistupujeme pouze k určitým datům tabulky
-

62. Popište metody, jak je možné udržovat denormalizovaný datový model v konzistentním stavu.

- Triggery
 - Uložené procedury
 - Aplikační logika
-

63. Popište co to je databázový pattern.

- odzkoušené a doporučené způsoby, jak řešit často se vyskytující požadavky
-

64. Jaké databázové patterny znáte?

- N-ární relace
 - Dědičnost
 - Přiřazení rolí
 - Klasifikace
-

65. Popište pattern přiřazení rolí. Jaké typy požadavků tento pattern řeší.

- Partneři kooperující s podnikem
 - škola – student, zaměstnanec, spolupracovník, přednášející, ...
 - podnik – zákazník, dodavatel, partner, zaměstnanec, ...
-

----- **TODO** -----

66. Popište nejjednodušší pattern přiřazení rolí. Popište jeho slabé a silné stránky.

- každá role je tvořena jinou entitou
 - silné: jednoduché, jasně definované role
 - slabé: není vhodný pro prostředí, kde často vznikají a zanikají role nebo se mění atributy.
-

66. Popište složitější patterny přiřazení rolí. Popište jejich slabé a silné stránky.

- silné: (umožňuje) odstranění redundance informací o osobách a organizacích
 - slabé: některá prostředí nejsou schopni rozlišit PARTY od rolí
-

67. Popište pattern klasifikace. Jaké typy požadavků tento pattern řeší.

- podpora členění instancí entity podle typů do kategorií a taxonomií

68. Popište nejjednodušší pattern klasifikace. Popište jeho slabé a silné stránky.

popis : veškeré info obsahuje jedna entita

silne:

Velice jednoduchý model, snadno pochopitelný pro všechny uživatele, Vhodný jako základ (prototyp), odrazový můstek pro pochopení a podrobnější analýzu, Implementace může používat omezení na hodnoty ve sloupcích nebo pouze uživatelská pravidla.

slabe:

-Velice nepružný model

 Přidání kategorie – přidání atributu

 Mnoho typů – mnoho atributů

-Více typů klasifikací – více sloupců (Product line 1, Product line 2);

.....

69. Popište složitější patery klasifikace. Popište jejich slabé a silné stránky.

.....

.....

70. Popište kritéria, které je nutné brát v úvahu pro výběr správného paternu.

Řešení musí odpovídat

- Složitosti business domény,
 - Složitosti business pravidel,
 - Schopnosti analytiků a vývojářů porozumět modelu,
 - Schopnosti uživatelů udržovat model.
-

71. Dimenzionální model slouží primárně:

- a) Pro analytické databáze
- b) Pro dohledové systémy

- c) Pro aplikace vyžadující zpracování velkého množství dat
- d) Pro aplikace vyžadující krátkou dobu odezvy

72. Mezi výhody dimenzionálních modelů patří

- a) Srozumitelnost pro koncové uživatele
- a) Jsou podporované analytickými nástroji
- b) Jsou snadno rozšiřitelné
- c) Snadná implementovatelnost

73. Popište pojem dimenze v dimenzionálním modelování.

.....

Dimensions are the foundation of the fact table, and is where the data for the fact table is collected. Typically dimensions are nouns like date, store, inventory etc. These dimensions are where all the data is stored. For example, the date dimension could contain data such as **year, month and weekday**.

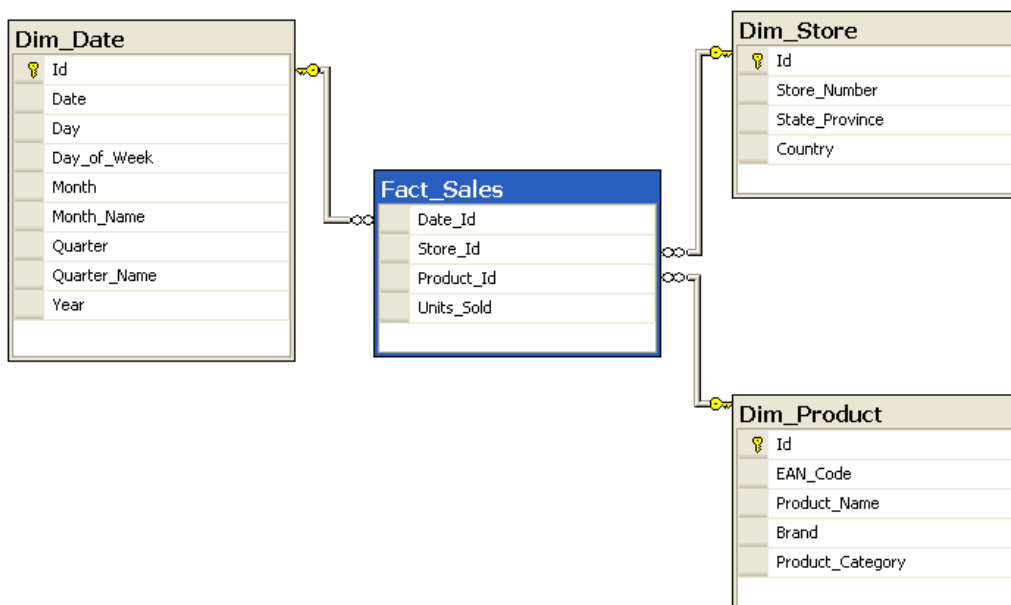
74. Popište pojem Faktové tabulky (metriky) v dimenzionálním modelování.

.....

most of the fact table rows are numerical, additive figures such as **quantity or cost per unit**, etc.

75. Popište Star schéma v dimenzionálním modelování.

Link -> [Wikipedia](https://en.wikipedia.org/wiki/Star_schema)



76. Kterých pět kroků vede k vytvoření dimenzionálního modelu.

- Výběr sledovaných procesů
- Určení granularity
- Určení dimenzí
- Určení metrik
- Definice získávání dat (ETL)

77. Který z uvedených typů není typ tabulky faktů?

- a) Transakce
- b) Průběžné transakce**
- c) Snapshoty
- d) Akumulující se snapshoty

78. Uveďte příklad aditivními, semiaditivními a neaditivní metriky v dimenzionálním modelu.

- additive = počet, cena v transakčních fact tabulkách
- semiadditive = počet, cena v snapshot tabulkách - součet za produkty má význam, za čas nemá význam
- nonadditive = procentuální profit

79. Uveďte základní tři typy dimenzí používané v dimenzionálních modelech.

- Statické
- Slowly growing
- Slowly changing dimension

80. Co to je slowly changing dimension?

- Při aktualizaci záznamů uvnitř DB mám 3 způsoby (typy), jak se s tímto problémem vypořádat

1. přepíši staré hodnoty za nové (staré hodnoty nikde nezalohuji a nejsou už dohledatelné)
2. přidám nový řádek do tabulky se stejným identifikátorem, jako má starší záznam.
3. uvádím expiration date jednotlivých záznamů

viz http://www.learn-datamodeling.com/sl_dim.htm

81. Co jsou hlavní cíle správy diskových prostorů RDMS?

- a) Udržet data stejného objektů blízko sebe.**
- b) Snížit počet nutných IO operací pro přenesení dat z disku do RAM**

c) Snížit opotřebení disků

d) Podpořit zabezpečení dat

e) Snížit nároky na administraci diskových prostorů

82. Jaké typy dat ukládá RDMS na disk? Uveďte alespoň čtyři:

- Data (tabulky, procedury)
 - Indexy
 - Transakční logy
 - Archivy databáze
 - Logy serveru
-

83. Na jaké menší logické a fyzické části se dělí datové soubory RDMS?

- data block
 - extent
 - segment
 - table space
-

84. Popište objekty Data block, Extent, Segment a Table space v databázi Oracle a vztahy mezi nimi.

-**Table space** - obsahuje několik segmentů, odpovídá logickému spojení objektů

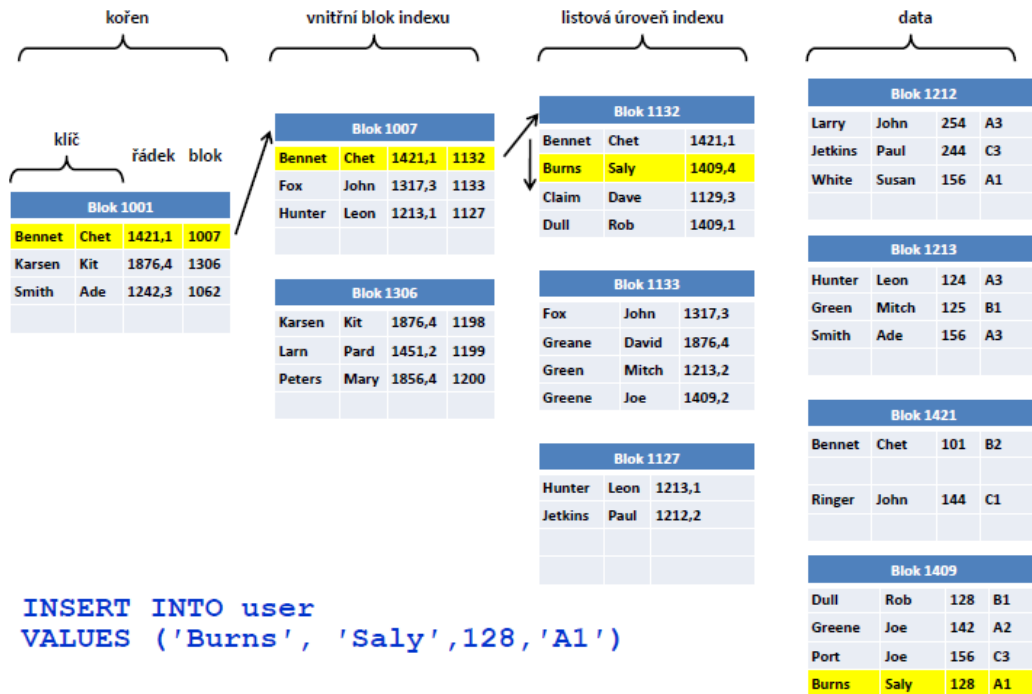
-**Segment** - obsahuje několik extentů, odpovídá tabulce či indexu

-**Extent** - obsahuje několik datových bloků, je to alokované místo pro konkrétní objekt

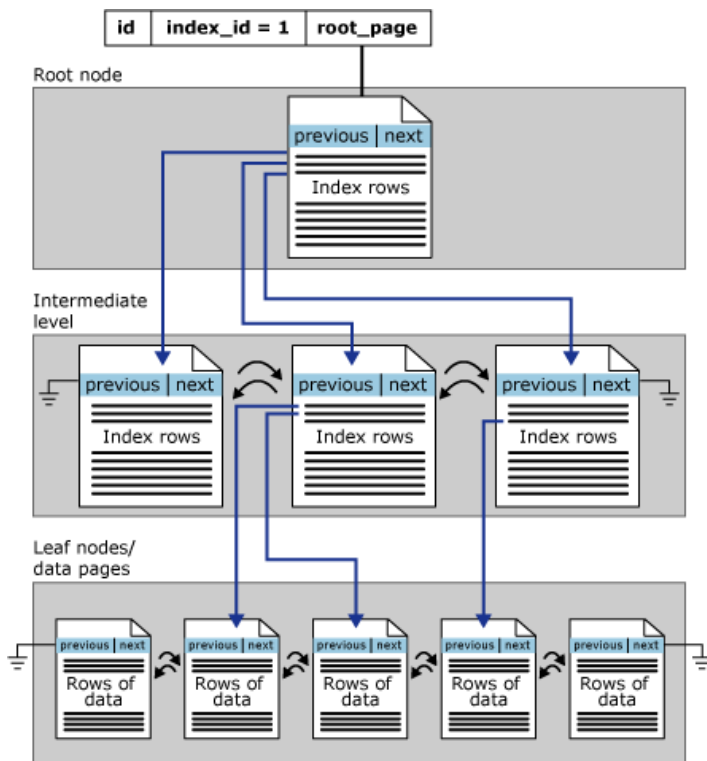
-**Data block** - obsahuje jednotlivé řádky dat

85. Popište (načrtněte) strukturu B-tree indexu help tabulky.

- odkaz od kořenového bloku na další bloky uvnitř DB. Postupně probublám až do listu stromu, kde jsou uloženy už jednotlivé řádky s daty viz. obrázek:



85. Popište (načrtněte) strukturu clustrované tabulky v systému Microsoft SQL Server.



86. Uvedte základní přístupové metody k datům help tabulky s indexem, které může použít RDMS.

Použití indexu

– Procházení indexu od kořene

where type in ('A','B')

– Nalezení hodnoty od kořene a scan nejnižší úrovně

where type > 'ABC'

– Scan nejnižší úrovně indexů

87. Popište strukturu bitmapového indexu a jeho použití.

relativně dobře vysvětleno na [wiki](#) nebo [zde](#)

- použití v případě, pokud má daný sloupec nízkou kardinalitu dat (např. pohlaví M/F)
 - použití hlavně pro read-only tabulky - přepočítání bitmap indexu je náročná operace
-

88. Proč je diskový subsystém kritickou oblastí pro výkon RDMS?

Disky jsou pomalé v porovnání se sběrnicemi a pamětí, každý přístup k disku je nákladný, je třeba minimalizovat počet přístupů na disky.

89. Vysvětlete zkratky SQL, DML, DDL, PL/SQL, Transact SQL.

SQL - Structured Query Language - jazyk pro práci s DB

DML – Data Manipulation Language – Insert/Update/Delete

DDL – Data Definition Language – Create/Drop/Grant/Revoke

PL/SQL (Procedural Language/Structured Query Language) je procedurální nadstavba jazyka SQL od firmy Oracle založená na programovacím jazyku Ada.

Transact-SQL (T-SQL) je proprietární rozšíření do SQL od společností Microsoft a Sybase. Microsoft tento jazyk používá v produktu Microsoft SQL Server. Sybase Software v Adaptive Server Enterprise.

90. Jaký je rozdíl mezi konceptem Identity a Sekvencí pro generování číselných řad.

Identity	Sequence
Table Specific	Table Independent
You cannot obtain the new value in your application before using it	You can obtain the new value before using it in an INSERT statement
You cannot add or remove the property from an existing column	You can add or remove a default constraint defined for a column with an expression that generates a new sequence value (extension to the standard)
You cannot generate new values in an UPDATE statement when needed, rather only in INSERT statements	You can generate new values in an UPDATE statement
The semantics of defining ordering in a multi-row insert are confusing, and in SELECT INTO statements are actually not guaranteed	The semantics of defining ordering in a multi-row insert are very clear using an OVER clause (extension to the standard), and are even allowed in SELECT INTO statements
You cannot define: minimum and maximum values, whether to allow	You can define minimum and maximum values, whether to allow cycling, and a cache size option for performance

cycling, and caching options	(extension to the standard)
You can reseed an identity property, but you cannot change the step size	You can alter any of the properties of a sequence object besides the data type, including the current value, increment, minimum value, maximum value, cycle and cache size
You cannot obtain a whole range of new identity values in one shot, letting the application assign the individual values	You can obtain a whole range of new sequence values in one shot using the stored procedure <code>sp_sequence_get_range</code> (extension to the standard), letting the application assign the individual values for increased performance

91. Mezi základní klauzule příkazu select nepatří:

- a) Group by
- b) Compute**
- c) Having
- d) Model by**
- e) Order by
- f) Where

92. Jaké typy joinů a jejich syntaxí znáte? Uveďte alespoň čtyři typy joinů.

- FULL Join
- CROSS Join
- INNER Join
- NATURAL JOIN
- OUTER Join [left|right] (Oracle (+))

93. Umožňuje select příkaz rekurzivní zpracování? Pokud ano, uveďte příklad.

Ano.

```
SELECT ROWNUM AS ID FROM dual CONNECT BY LEVEL <= 10
```

94. jaký je rozdíl mezi použitím WITH klauzulí a odvozených tabulek v SELECT příkazu?

WITH dovoluje v dotazu specifikovat "poddotazy" a ty nejak pojmenovat (WITH neco AS (SELECT blabla)).

Odvozene tabulky ... ?

Zdroj: [link](#)

95. Co to jsou agregační funkce v příkazu select? Uveďte příklad.

Agregační funkce jsou v SQL statistické funkce, pomocí kterých systém řízení báze dat umožňuje seskupit

vybrané řádky dotazu (získané příkazem SELECT) a spočítat nad nimi výsledek určité aritmetické nebo statistické funkce. Agregiční funkce se v SQL používají s konstrukcí GROUP BY.

select count(*) from title -- Počet řádků.

select count(price) from title -- Počet nenulových hodnot price.

select count(distinct price) from title -- Počet různých nenulových hodnot price.

96. Co to jsou analytické funkce v příkazu select? Uveďte příklad.

Různé statistické funkce - například RANK() nebo LAG():

```
SELECT country_region REGION,  
        country_name KRAJINA,  
        SUM(amount_sold) PREDAJ,  
        RANK() OVER (  
            PARTITION BY country_region  
            ORDER BY SUM(amount_sold) DESC NULLS LAST  
        ) AS PORADIE,  
        LAG(SUM(amount_sold),1) OVER (  
            PARTITION BY country_region  
            ORDER BY SUM(amount_sold) DESC  
        ) - SUM(amount_sold) ROZDIEL  
FROM sales, customers, countries  
WHERE sales.cust_id = customers.cust_id AND  
        customers.country_id = countries.country_id AND  
        sales.time_id BETWEEN  
            TO_DATE('20001101','yyyymmdd') AND  
            TO_DATE('20001130','yyyymmdd')  
GROUP BY country_name, country_region;
```

97. Obsahuje příkaz update klauzuli from?

Ano, může.

```
UPDATE employee_bonus SET bonus = 0 FROM employee_bonus b INNER JOIN employees e
```

IN b.employee_id = e.employee_id **WHERE** e.bonus_eligible = 'N'

98. Udává syntaxe ANSI INNER JOIN pořadí tabulek, v jakém se spojují?

Ano

(zdroj: MI-DSP, přednáška 7 slide 38)

99. Jak se dělí funkce definované v databázi? Uveďte alespoň jedno dělení.

Například podle datového typu:

- číselné
 - práce s řetězci
 - práce s časem
 - konverzní funkce
-

100. Co to je databázový katalog?

Shromažďuje statistické informace pro výpočet cen plánů pro dotazy.

101. Jaký je vztah mezi skriptem, dávkou (batch) a příkazem při psaní kódu v RDMS?

.....
.....
.....

102. Jaké typy triggřů znáte?

- DML triggery.
 - Triggery na úrovni DB.
-

103. Uveďte alespoň dva důvody, proč není vhodné používat triggery.

- Špatné ladění, možný zdroj chyb.
 - Snižují výkon.
 - nejsou v systému vidět
-

104. Z jakých důvodů je vhodné použít trigger?

- a) Ověření integritních omezení a složitějších business pravidel
 - b) Implementace logování a auditu
 - c) Implementace business logiky
-

106. Jaké typy chyb předává RDMS klientům?

Chyby na úrovni

- Hardware
 - Software
 - Aplikace
 - Kódu
-

107. Jaké různé výsledné stavy z pohledu aplikace nebo klienta mohou být po výskytu chyby v RDMS?

Kód je ukončen, transakce je ukončena

Kód je ukončen, transakce pokračuje

Kód pokračuje, transakce ukončena

Kód pokračuje, transakce pokračuje

108. Uveďte některé dobré a špatné praktiky při psaní kódu v RDMS.

Dobré praktiky:

- Podpora kompilátoru
 - Přímé předávání parametrů
 - Konstanty jako proměnné
 - Transakční uzávěr
 - Přebrání rozhodování při selektech typu search do kódu
 - Rozumná hierarchie view
- Porovnávat/dosazovat vždy stejné typy
- Používání exists vždy, když to jde
- Set nocount on na začátku každé procedury
- Nezapomínat na hodnotu null
- Omezovat zbytečné testy pro řízení toku zpracování

Špatné praktiky:

- Ukládání ROWID pro pozdější reference
- Použití sekvencí jako počítadla
- Nahrazení null hodnotami jako „N/A“ nebo „Unknown“
- Použití select * from
- Použití výrazů v join klausuli

- Použití identifikátorů vyžadující uvozovky

109. Co to je null v relačním modelu? Jaké komplikace přináší null v SQL a při psaní kódu?

- NULL is the value used to represent an unknown piece of data.

- je třeba speciálně ošetřit

-nepoužívat null spolu s operátorem porovnání =, používat IS, = se chová různě dle kontextu

110. Uveďte vhodný a nevhodný příklad pro použití view při návrhu databáze a psaní kódu.

(+) - Typickým příkladem může být např. práce s tabulkou Zaměstnanců, která mimo jiné obsahuje info o rodných číslech. Abych nemusel konkrétně nastavovat práva na čtení, vytvořím v DB VIEW odpovídající této tabulce ovšem bez sloupce RČ. Uživatelé pak budou používat už pouze toto VIEW.

.....

111. Co znamená zkratka ACID?

Databázové transakce musí splňovat tzv. vlastnosti ACID:

Atomicita

Databázová transakce je jako operace dále nedělitelná (atomární). Provede se buď jako celek, nebo se neprovede vůbec (a daný databázový systém to dá uživateli na vědomí, např. chybovou hláškou).

Konzistence

Při a po provedení transakce není porušeno žádné integritní omezení.

Izolovanost

Operace uvnitř transakce jsou skryty před vnějšími operacemi. Vrácením transakce (ROLLBACK) není zasažena jiná transakce, jinak i tato musí být vrácena. V důsledku tohoto chování může dojít k tzv. řetězovému vrácení (cascading rollback).

Trvalost

Změny, které se provedou jako výsledek úspěšných transakcí, jsou skutečně uloženy v databázi a již nemohou být ztraceny.

112. K čemu slouží savepoint? Popište jeho použití.

A savepoint is a way of implementing subtransactions (also known as nested transactions) within a relational database management system by indicating a point within a transaction that can be "rolled back to" without affecting any work done in the transaction before the savepoint was created. Multiple savepoints can exist within a single transaction. Savepoints are useful for implementing complex error recovery in database applications — if an error occurs in the midst of a multiple-statement transaction, the application may be able to recover from the error (by rolling back to a savepoint) without needing to abort the entire transaction.

113. Popište chained a unchained mód.

Chained:

- Začátek transakce
 - První příkaz (ne každý)
 - Insert, update, delete, select for update/holdlock, ...
- Konec transakce
 - Commit, rollback

Unchained:

- Každý příkaz autonomní transakce
- Začátek transakce
 - Explicitně begin tran
- Konec transakce
 - Commit, rollback

114. Co to jsou izolační úrovně? Jaký je rozdíl mezi úrovní Read committed a Serializable?

Určuje "úroveň zamčení" - jak moc bude databáze zamykat data během transakce.

Serializable -> [Wikipedia](#) (nejvyšší úroveň izolace, zámky na READ i WRITE, uvolněny na konci transakce)

Read committed -> [Wikipedia](#) (nižší úroveň izolace, zámky na WRITE uvolněny na konci transakce; zámky na READ jsou uvolněny hned, jakmile doběhne SELECT)

(zdroj: [Wikipedia](#))

115. Popište optimistické a pesimistické schéma zamykání.

In pessimistic locking a record or page is locked immediately when the lock is requested, while in an optimistic lock the record or page is only locked when the changes made to that record are updated.

116. Co to je deadlock? Dá se deadlockům zabránit? Jak se dá deadlockům předcházet?

- situace kdy si dvě transakce navzájem blokují své zdroje a čekají na jejich uvolnění. Transakce si však nemohou dát přednost a proto je třeba nějaké vyšší autority jako např. dat. serveru.

- databazové servery dokáží deadlocky identifikovat (např. se zruší jedna transakce)

-předchází se např. používání krátkých transakcí, přístupováním na zdroje ve stejném pořadí, co nejnižší granularitou zamykání nebo porušením jedné z [Coffmanových podmínek](#).

117. Napište hlavní oblasti činností při administraci RDMS.

- Instalace.
 - Zálohování.
 - Dostupnost.
 - Sledování.
 - Ladění.
 - Zajišťování bezpečnosti.
 - Řešení plánovaných i neplánovaných problémů.
-

118. Definujte cíle zálohování RDMS.

- Podpora zvýšení dostupnosti - co nejrychlejší obnova do posledního možného stavu.
 - Schopnost vrátit se k libovolnému stavu v minulosti (eliminace lidských chyb).
-

120. Popište alespoň tři zálohovací metody RDMS.

- Kopie všech diskových prostorů.
 - Export zálohovaných dat.
 - Záloha databáze pomocí prostředků serveru.
-

119. Definujte cíle zvyšování dostupnosti RDMS.

.....
.....

121. Popište alespoň jeden způsob pro zvýšení dostupnosti RDMS.

- Zálohování.
 - Použití clusterů.
 - Mirroring.
-

122. Popište co to je near-online databáze.

Databáze udržovaná ve stavu blízkém produkci, replikované DB, klony diskových prostorů

123. Jaké typy replikací znáte?

- Multimaster.
 - Snapshot.
-

124. Jaké jsou cíle při sledování provozu RDMS administrátory?

- Předcházení nepředvídaným stavům.
 - Schopnost predikce změn požadavků.
 - Proaktivní řešení situací, jež mohou způsobit nedostupnost služby.
-

125. Jaké mají možnosti administrátoři při ladění výkonu?

1. Nastavení parametrů OS

parametry jádra, patche, velikost paměti, distribuce disků, I/O kanály

2. Nastavení parametrů datového serveru

stovky parametrů, nutný restart, dokumentace změn

3. Přidávání a rušení indexů

nejčastější metoda, přidání ale zamyká tabulky - dopad na provoz, mnoho indexů zpomaluje updaty, nároky na uložení

4. Používání speciálních výpočetních plánů

pro konkrétní příkazy konkrétní plány a to bez zásahu do kódu, plány se přiřazují na základě textu příkazu

126. Co to je autonomní transakce? K čemu slouží a jaké má její použití rizika?

Autonomní transakce: možnost pustit další transakci z jiné, nadřazené transakce.

Autonomní transakce začínají z *nadřazené* neboli *hlavní* transakce, ale pracují samostatně bez transakčního dohledu z dané nadřazené transakce. Jestliže v autonomní či v hlavní transakci použijeme potvrzení nebo odvolání či dojde-li z libovolného důvodu k chybě, druhou transakci to neovlivní.

S oblibou používáme tuto vymoženost na zápis do protokolu aplikačních událostí. Jestliže máme monitorovat aktivitu nezávisle na jejím výsledku a zároveň nemá úspěch či selhání zápisu do protokolu ovlivnit aplikaci samotnou, jsou autonomní transakce dokonalým řešením.

Užití: Logování i rollbacknutých transakcí?

127. Co to je Data profiling a k čemu se používá?

Úprava dat za účelem zvýšení výkonu či rozšíření funkcionality.

já zas našel něco jiného http://en.wikipedia.org/wiki/Data_profiling podle toho jde o analýzu dat ne úpravu, sloužící k lepší práci s daty (viz odkaz)

jo, máš pravdu, ne úpravu dat samotných ale metadat (upravu ci vytvoreni, pokud nexistujou)

128. Jaké jsou základní nutné schopnosti pro práci s daty podle metodiky Information Capability Framework? (šest schopností)

Describe, Organize, Integrate, Share, Govern, Implement (DOInSGIm)

zdroj: přednáška 1. Sprava dat v podniku - slajd 14

129. Uveďte alespoň čtyři příklady dimenzí datové kvality.

Timeliness and Availability

Consistency (Completeness)

Accuracy

Duplication

Data Specifications

130. K čemu slouží popisy datových toků v datově orientovaných systémech organizace?

K vizualizaci, co je *vstupem/výstupem* jednotlivých komponent, *kdo* komunikuje s *kým*, *kde* je co uloženo.

131. Jaké typy analýz metadat znáte? K čemu jednotlivé typy analýz slouží?

Dopadová analýza (Pokud místo Y/N začneme používat A/N, co všechno musíme zkontrolovat?)

Lineage analýza - Upstream (kdo si všechno bere data odnekud?) / Downstream (jaká data se podílejí na hodnocení?)

132. Uveďte alespoň dva přístupy a dvě technologie používané pro indikaci změn v datech v rámci integrace.

za mě:

- Timestamp

- Fronta událostí

- Technologicky (triggery)

- Aplikačně

zdroj: přednáška 2. Integrace dat - slajd 18

133. Kdo určuje kódovou stránku textů při komunikaci mezi klientem a databázovým serverem?

a) Klient

b) Datový server

c) Operační systém klienta

d) Operační systém datového serveru

e) Uživatel

134. Co to je a k čemu slouží procesní matice při návrhu datového modelu?

Tipnul bych si dle přednášky 4, slide 57, že jde o tabulku, která ukazuje jak náročné jsou určité dotazy a jak frekventované.

Slouží k identifikaci primárních klíčů a dalších indexů při tvorbě fyzického modelu

Tiež na optimalizáciu určených joinov.

135. (ve skutečnosti to je číslo 38.): Které jsou nejdůležitější požadavky na systémy pro datovou integraci?

o Stabilita

o Udržovatelnost

o Modifikovatelnost

o Správa a dohled

o Škálovatelnost

o Způsob vývoje

o Úplnost

o Otevřenost

o Podpora

136. (ve skutečnosti to je číslo 39.): Popište, čím se liší MDM koncepty Master Reference data a Master System of Records.

TODO

In computing, Master Data Management (MDM) comprises a set of processes, governance, policies, standards and tools that consistently defines and manages the master data (i.e. non-transactional data entities) of an organization (which may include **reference data**).

Reference data: data from outside the organisation (often from standards organisations) which is, apart from occasional revisions, static. (currency codes, Countries)

Master Data: is also relatively static data but originating from within the organisation e.g. products, departments, even customers.

A **system of record** (SOR) or Source System of Record (SSoR) is Data Management term for an information storage system (commonly implemented on a computer system) that is the authoritative data source for a given data element or piece of information. The need to identify systems of record can become acute in organizations where management information systems have been built by taking output data from multiple source systems, re-processing this data, and then re-presenting the result for a new business use.

In these cases, multiple information systems may disagree about the same piece of information. These disagreements may stem from semantic differences, differences in the timing of the ETL extracts that create the data they report against, or may simply be the result of bugs.

137. (ve skutečnosti to je číslo 47.) Co to je MDM. K čemu v organizaci slouží a proč je důležité?

In computing, Master Data Management (MDM) comprises a set of processes, governance, policies, standards and tools that consistently defines and manages the master data (i.e. non-transactional data entities) of an organization (which may include **reference data**).

Master data: This key business information may include data about customers, products, employees, materials, suppliers, and the like.

MDM is supported by removing duplicates, standardizing data (Mass Maintaining), incorporating rules to eliminate incorrect data from entering the system in order to create an authoritative source of master data.

MDM has the objective of providing processes for collecting, aggregating, matching, consolidating, quality-assuring, persisting and distributing such data throughout an organization to ensure consistency and control in the ongoing maintenance and application use of this information.

At a basic level, MDM seeks to ensure that an organization does not use multiple (potentially inconsistent) versions of the same master data in different parts of its operations, which can occur in large organizations.

One of the most common reasons some large corporations experience massive issues with MDM is growth through **mergers** or **acquisitions**. Two organizations which merge will typically create an entity with duplicate master data (since each likely had at least one master database of its own prior to the merger). Ideally, **database administrators** resolve this problem through **deduplication** of the master data as part of the merger. In practice, however, reconciling several master data systems can present difficulties because of the dependencies that existing applications have on the master databases. As a result, more often than not the two systems do not fully merge, but remain separate, with a special reconciliation process defined that ensures consistency between the data stored in the two systems. Over time, however, as further mergers and acquisitions occur, the problem multiplies, more and more master databases appear, and data-reconciliation processes become extremely complex, and consequently unmanageable and unreliable. Because of this trend, one can find organizations with 10, 15, or even as many as 100 separate, poorly integrated master databases, which can cause serious operational problems in the areas of customer satisfaction, operational efficiency, decision-support, and regulatory compliance.

2014 další otázky

135. Co to "Information Capability Framework" a které základní schopnosti jsou nutné pro správu a využití dat

The **information capabilities framework** is the people-, process- and technology-agnostic set of capabilities needed to describe, organize, integrate, share and govern an organization's information assets in an application-independent manner in support of its enterprise information management (EIM) goals.

Zdroj: <http://www.gartner.com/it-glossary/information-capabilities-framework>

136. Jak se pozná, že jsou data nekvalitní?

IMHO: Nesplňují technické a významové požadavky na kvalitu dat

137. Kdy a jak vzniká nekvalita dat?

- Při zadání dat
- Při předávání dat mezi systémy
- Časem

(Zdroj: MI-DSP, Přednáška 2, slide 20)

138. Kdo a jak pozná, že jsou data nekvalitní? Uveďte příklady nekvality dat.

.....

139. Co to je Buss Matrix, k čemu slouží?

Matica: business procesy vs. dimenzie

Služi k vyberu sledovanych procesov a dimenzii k nim prisluchajucich/pozadovanych.

140. Co to je profiling dat? K čemu se používá?

(viz 127.)

141. Jak se dá prokázat, že jsou informace získané z dat kvalitní?

IMHO: Splňují technické a významové požadavky na kvalitu dat

142. Co to jsou byznys metadata? K čemu slouží?

- Jednotný slovník organizace
- Komunikace
 - Mezi odděleními
 - Mezi Byznysem a IT
 - Řešení výjimek

- Požadavky
 - Schvalovací proces
 - Diskuse
 - Více druhů slovníků

(Zdroj: MI-DSP, Přednáška 3, Slide 7)

143. Jak se liší byznys metadata od technických metadat?

Byznys metadata

- viz výše

Fyzická metadata

- Popisy datových modelů
 - Logická úroveň – jednotný model organizace
 - Fyzická úroveň – modely jednotlivých databází
- Popisy reportů
 - Jaká data se používají
 - SQL dotazy
 - Kdo a kdy je používá
 - Popis na byznys úrovni
- Popisy transformací
 - Zdroje a cíle
 - Transformační pravidla

144. Co jsou zdroje technických metadat?

- Modelovací nástroje
 - Logické modely
 - Fyzické modely
 - Mapování a transformace
- Databáze
 - Fyzické modely
 - Skripty s transformacemi
- ETL nástroje
 - Transformace
- Reportingové nástroje
 - Zdroje dat
 - Univerzum
 - Transformace a výpočty

(Zdroj: MI-DSP, Přednáška 3, slide 9)

145. Co jsou operační metadata a k čemu slouží?

- jedná se o popis jednotlivých výsledků během různých operací v datovém skladu
- příkladem může být ETL proces kdy pomocí operačních metadat je zaznamenáván čas spuštění, čas ukončení, s jakými daty bylo manipulováno, atd.
- zachycují jak a kdy je datový sklad aktualizován a využíván

(Zdroj: <http://czm.fel.cvut.cz/vyuka/A4M33CPM/Download/Metadata.pdf>, slide 5)

146. Jaké typy analýzy metadat se používají?

- Data Lineage
 - Upstream
Které aplikace používají centrálních číselník měn?
 - Downstream
Která všechna data se podílejí na ohodnocení spolehlivosti dodavatele?
- Impact analysis
 - Které všechny tabulky a aplikace se budou muset upravit, když přejdeme z kódování ISO88592 na kódování UTF8?
 - Pokud místo Y/N začneme používat A/N, co všechno musíme zkontrolovat?

(Zdroj: MI-DSP, Přednáška 3, slide 14)

147. Jaký je rozdíl mezi synchronním a asynchronním předáváním dat?

Asynchronní

- V jednom okamžiku mají různé systémy různá data
- Technologicky jednodušší
- Nižší požadavky na průchodnost systému
- Messaging

Synchronní

- Zaručuje konzistentní stav ve všech systémech pro všechny uživatele
- Výpadek jednoho systému ovlivňuje všechny ostatní
- Dvojfázový commit

(Zdroj: MI-DSP, Přednáška 4, slide 10)

148. Jaký je rozdíl mezi Federativním a Mediativním přístupem k integraci dat?

Federation

- Systém umožňuje (vynucuje) aby požadavky vznikaly jeho prostřednictvím a rozprostírá je do jednotlivých systémů.
- MDM aplikace

Mediation

- Reaguje se na změny v jednotlivých systémech a ty se předávají ostatním systémům
- Messaging
- Replikace

(Zdroj: MI-DSP, Přednáška 4, slide 12)

149. Jaký je rozdíl mezi Point-to-point a Hub-and-spoke integračním modelem?

.....
.....

150. Co to je denormalizace?

Opačný proces k normalizaci. Porušení normálních forem úpravami tabulek například kvůli zvýšení výkonnosti.

151. Jaké techniky se používají při indikaci dat, které je nutno přenášet v rámci integrace?

.....

.....

152. Jaké jsou hlavní problémy při vzniku nového záznamu v integračním systému?

- Neúplný záznam
- Nekonzistentní záznam
- Duplicitní záznam

(Zdroj: MI-DSP, Přednáška 4, slide 17)

153. Jaké jsou hlavní problémy při změně záznamu v integračním systému?

- Porušení konzistence
- Rozpoznání nezměněné položky
- Vytvoření duplicity, neúplného záznamu

(Zdroj: MI-DSP, Přednáška 4, slide 18)

154. Jaké jsou hlavní problémy při zrušení záznamu v integračním systému?

.....

.....

155. Jak se používá datová kvalita při integraci dat z více systémů?

.....

.....

156. Jaké architektury MDM se používají?

MOŽNÁ TOTO:

Collaborative style

Operational style

Analytical style

Zdroj: <http://www.ibm.com/developerworks/data/library/techarticle/dm-0703sauter/>

157. Které aktivity je nutné provést při převodu logického datového modelu na fyzický datový model?

.....

.....

158. Jaká jsou hlavní rizika integračních projektů?

- o Bezpečnost
 - ztráta informací
 - neautorizované modifikace
 - právní odpovědnost
 - pravdivost informací
 - původ informací
 - krádež služeb
 - ztráta důvěry zákazníků
 - příležitost pro fraud

(Zdroj: MI-DSP, Přednáška 4, slide 34)

159. Které všechny služby centralizuje architektura Klient-server?

- Centralizace
 - Modelu
 - Dat
 - Byznys logiky – funkcionality
 - Integrity - ověřování dat
 - Ověření dat na klientovi
 - Duplikace ověření
 - Bezpečnostních pravidel
 - Autentizace na úrovni server - aplikace
 - Autorizace na úrovni aplikace
 - Role na úrovni serveru
 - Nasazení změn
 - Nutnost podporovat více verzí klientů

(Zdroj: MI-DSP, Přednáška 5, slide 31)

160. Jaké jsou základní kroky životního cyklu dotazu v DBMS?

1. Navázání spojení s klientem
2. Porozumění požadavku
3. Optimalizace a vytvoření výpočtu
4. Vlastní výpočet
5. Předání výsledků

(Zdroj: MI-DSP, Přednáška 5, slide 34)

161. Jak probíhá navázání spojení mezi klientem a serverem?

- Klientský software
- Informace o uživateli – identifikace, časové pásmo, kódová stránka
- Vytvoření klientského procesu
- Adresářové služby pro nalezení serveru
- Síťový listener
- Navázání spojení
- Vytvoření procesu spravující klientské připojení
- Alokace struktur pro správu klientského připojení
 - Síťová komunikace

- Prostor pro výsledky
- Lokální prostor pro výpočty
- Prostor pro uživatelská data

(Zdroj: MI-DSP, Přednáška 5, slide 35)

162. Co to je konceptuální model a co obsahuje?

.....
.....

163. Co je logický datový model a co obsahuje?

.....
.....

164. Jaké aktivity je nutné provést při převodu konceptuálního datového modelu na logický datový model?

.....
.....

165. Jaké jsou hlavní rozdíly mezi relačním a objektivě orientovaným modelováním?

.....
.....