

- datové sklady - metody uskladnění**  
1) **MOLAP** (Multidimenz. OLAP) – vícerozměrný spreadsheet, datová krychle, hierarchické dimenze, schéma hvězdicové či vločkové; 2) **ROLAP** (Relační OLAP) – speciálně navržená relační databáze (struktura propojených tabulek), pomalejší než MOLAP;
- data mining - metody dolování dat**  
Rozhodovací stromy, lineární/logistická regrese, diskriminační/seskupovací/faktorová analýza, met. hledání asociačních pravidel, met. nejbližších sousedů, met. sdružování modelů, neuronové sítě, Bayesovské metody (Bayesovská síť)
- napište kdy a pro jaké operace může být definován**  
kdv: před provedením akce – BEFORE, po provedení akce – AFTER, při splnění podmínky – WHEN (podmínka); pro operace: INSERT, UPDATE, DELETE
- nad kterými pohledy lze provádět DML dotazy**  
nad materializovanými pohledy (update, delete, insert), nebo nad jednoduchými nematerializovanými (nad jednou tabulkou)
- R, OO, O-R DB - porovnejte je a napište charakteristiku**  
R = relační: starší Oracle do verze 7, DB2, SQL-92; OO = objektově orientované: akademická půda, Manifest OODS98, ODL, OQL, objekty s UID; O-R = objektově-relační: Oracle od verze 8, Informix – využívané v praxi, SQL3 (SQL99), naproti „R“ navíc abstraktní datové typy;
- RBO - co to je. Kdy se to nechová optimálně. –**  
Je to: = Rule Based Opt., starší přístup optimalizace podle syntaxe příkazu a existence indexů. Dnes Oracle používá implicitně CBO (Cost Based Opt.); Neoptimální chování: Při výběrání indexu z tabulky, existuje-li v ní více neunikátních indexů;
- temporální databáze - lze realizovat ideální model? Proč?**  
Nelze: Dosažení ideálního modelu je prakticky nemožné. Protože by nutné uchovávat celou souvislou historii dat v DB.
- které datové struktury lze použít pro uložení objemných dat – LOB (Large Object): 1) externí: BFILE (jen čtení); 2) interní: BLOB (Binary LOB), CLOB (Character LOB), NLOB (Nation. CLOB); max. velikost 4GB**
- OLTP a OLAP rozdíl**  
OLTP = Online Transaction Processing Systém, DB systém, orient. na zákazníka, současná data, ER schéma, velikost DB až GB (normalizované tabulky); OLAP = Online Analytical Processing, DataWarehouse, orientovaný na trh, historická data, agregovaná data, velikost DB až TB (redundantní data);
- Charakteristiky distribuované DB**  
výhody: lok. autonomie, lepší výkon, spolehlivost (výpadek PC neohrozí DB), rozšiřitelnost a schopnost sdílet informace; nevýhody: složitost, cena, bezpečnost, obtížný přechod;
- Jaké příkazy SQL se mohou používat v Pg/SQL bez omezení a jaké s omezením.**  
s omezením: INSERT, UPDATE, DELETE – nelze používat na pohledy, používá se proto INSTEAD OF (ale i to je omezená, nelze množ. operátory, skup. operace, join, distinct); bez omezení: SELECT – v potřebných situacích se vytvoří implicitní kurzor.
- Co je to ODBC.**  
= Open Database Connectivity, specifikace API nezávislé na DB a jejím jazyku; DriverManager poskytuje příslušné DB závislé ovladače;
- Definujte semistrukturovaná data.**  
Neuspořádaná či neúplná data, jejíž struktura se může měnit, dokonce i nepredikovatelným způsobem. XML je instancí semistrukturovaných dat. Dále např. HTML stránky, biologická data.
- Naspat možnosti, jak lze v Pg/SQL dělat cykly.**  
1) LOOP tělo END LOOP; 2) WHILE podm. LOOP tělo END LOOP; 3) FOR prom. IN [REVERSE] start...end LOOP tělo END LOOP;
- Vysvětlit funkci trigger.**  
trigger = událostně spouštěná procedura PL/SQL spojená s tabulkou; činná DB aktivní: zajišťuje řízení transakcí, bezpečnost, ref. integ., aplikační pravidla, audit, synchronizaci tab. a statistiku;
- Podmínky bezpečnosti vyhodnocení Datalogových pravidel**  
proměnná X (buď v hlavě, v negovaném podcíli, nebo v porovnávacím predikátu) se musí vyskytovat také v normálním pozitivním podcíli těla; (bezpečnost = smysluplnost)
- Definujte v SQL99 řádkový typ a použijte jej v definici tabulky**  
definice řádk. typu: CREATE ROW TYPE typ\_adresa (ulice CHAR VARYING(50), město CHAR VARYING(35)); definice tabulky: CREATE TABLE osoba (jmeno CHAR VARYING(50), adresa typ\_adresa);
- Vysvětlit GROUP BY CUBE**  
GROUP BY s1 CUBE s2 vytvoří dimenzi (drill-down) na proměnných z agregovaných seznamů (GROUP BY);
- Na jaké typy se člení distr. DBS podle hlediska autonomie (=izolovanost) lokálních systémů?**  
těsně integrované – uživatel vidí data centralizovaná v jedné DB, každý lok. sys. má úplnou znalost o datech v celém DBS; semiautonomní – sdílená jen část dat; zcela autonomní – izolované, lokální sys. nezávislé, neví o ostatních DBMS;
- K čemu slouží JDBC? Jeho princip?**  
slouží jako Java rozhraní k unifikovanému přístupu k datům (v DB či souborech); princip: Využívá dodané ovladače pro konkrétní SRBD zastíněné rozhraní JDBC (java.sql.\*, javax.sql.\*);
- Čím je definováno XML?**  
obecně: logickým vyznačováním pomocí elementů (s atributy) v stromové struktuře s jedním kořenovým elementem; konkrétně: specifikací struktury pomocí šablony DTD (Document Type Definition), XSD, Relax či jinými;
- K čemu slouží nepojmenovaný PL/SQL blok?**  
K použití v místě definice, např. v těle triggeru, nebo v SQL dávce;
- Definujte odvozenou horizontální formu.**  
= fragmentace tabulky horizontálním řezem na dvě a více části na základě relace s jinou tabulkou;
- Napište definici typu tabulky zákazníka a typu adresa v SQL99.**  
CREATE TABLE Zakaznik (adresa ROW (ulice VARYING CHAR(50), mesto VARYING CHAR(35));
- Co musí splňovat podmínka vyhodnotelnosti, jeli zapsána v negované formě?**  
zápis pravidel musí být bezpečný, tj. v zápisu se podmínka musí vyskytovat v normálním pozitivní formě (pozitivní podíl těla podmínky);
- Popište běžné OLAP operace.**  
Slice: jedna dimenze kostky se stanoví napevno a udělá se řez; Dice: slice (řez) 2 a více dimenzemi datové kostky; Drill Down/Up: zpřesnění (down) nebo zobecnění (up) u hierarchií (např. down je kraj -> město -> ulice); Roll-up: počítá různé součty dat příbuzných pro jednu nebo více pevně stanovených dimenzí; Pivot: změna orientace kostky (tzn. změna pohledu na data);
- Kam mohou být mapována integritní omezení (zajistí referenční integrity, kardinality vazeb, doménové integrity atd.) v architektuře klient/server.**  
1) na server jako součást DDL či ve formě uložených procedur; 2) na klienta jako součást zpracování (obtížná udržitelnost)
- Který dokument (XML) se nazývá "dobře vytvořený".**  
tzv. well-formed, tj. splňuje 7 základních pravidel, např. párové tagy, správné hnždění elementů, nehnždění komentářů; well-formed neznamená validní (podle DTD, XSD, RelaxNG);
- Dana transakční databáze: T1={A,B,C}, T2={A,C,D}, T3={C}. Napíšte silná asociativní pravidla (včetně jejich support a confidence) při s=50% a c=70%**
- Které z dokumentů D1...D4 budou vybrány jako odpověď na dotaz Q?**  
Budou vybrány ty dokumenty, u nichž se shodují všechny jedničky vůči Q;
- Jake typy tabulek obsahuje hvězdicové schéma a jaky je jejich ucel (co je obsahem)**  
1) tab. faktů; 2) tab. dimenzí; 3) tab. měřtek (např. Kč, cm); jsou zde uloženy data či skutečnosti.
- Jake prostředí má ODL k vyjádření dědičnosti**  
UNDER: klíčové slovo specifikace dědičnosti typu od typu INSTANTIABLE: modifikátor fce vyjadřující absenci její implementaci; OVERRIDING: modif. fce vyjadř. předefinování metody předka; FINAL: modif. fce vyjadř. nepředefinovatelnost v potomcích;
- Dobře vlastnosti XML: validní, strukturovaný, možnost vyhledávání**
- příklad: vytvořit abstraktní datový typ Stipendium**  
CREATE TYP Stipendium AS (navez VARYING CHAR(20), INSTANTIABLE NOT FINAL METHOD cena() RETURNS DECIMAL);
- Je zadána extenzionální databáze q(X,Y) a r(W,X,Y) a intenzionální pomocí pravidla p(X,Y) :- q(X,Y), not(r(W,X,Y)), X < Y**  
**Vyjmenujte důvody proč pravidlo p není bezpečné.**  
prom. W se vyskytuje v negovaném podcíli, ale nevyskytuje se v normálním pozitivním podcíli těla;
- Objasněte na co se v SQL používá ON DELETE CASCADE.**  
Definuje se pro tabulky kaskádně závislé na nadřazených tabulkách. Je-li smazán referovaný záznam z nadřazené tabulky, jsou odstraněny také závislé záznamy v této tabulce;
- Jak byste implementovali přírůstek v architektuře klient -- server?**  
implementace části na serveru, části na klientu, testování a instalace přírůstku;
- Co je to DML, DDL a DCL? (Jaké mají funkce?)**  
DML: Data Manipulation Language, manipuluje s daty, prostředky pro popis algoritmů, transformace stavu DB, CRUM; DDL: Data Definition Language – definice a popis dat, pravidla pro jejich uložení a přístup k nim, akce při porušení sémantické integrity; DCL: Data Control Language – řízení dat, ochrana přístupu k datům;
- Vyjmenujte všechny způsoby vkládání hodnot do tabulky v SQL. Uveďte příklady.**  
1) příkazem INSERT (v dávce, proceduře, fci, triggeru), vkládání hodnot v nových záznamech, příklad: INSERT INTO tab (s1, s2) VALUES (jedna', dva'); 2) příkazem UPDATE (v dávce, proceduře, fci, triggeru), vkládání hodnot do již existujících záznamů, příklad: UPDATE TABLE tab SET s1='tři' WHERE s1='jedna'; 3) pomocí TRIGGERŮ referencováním na pomocí „new.“ a přepisováním hodnot; příklad: :new.s1 = „tři“;
- Jak lze v SQL realizovat entitní a doménové omezení. Uveďte příklady.**  
entitní omezení: zajištěním jednoznačné identifikace každého řádku relace pomocí jednozn. primárního klíče, př.: CREATE TABLE ( ... PRIMARY KEY (s1) ); doménové omezení: zajištění, aby každá hodnota atributu byla nezaměnitelná (CREATE DOMAIN Halere NUMERIC(12,2) CONSTRAINT padesatnik CHECK (value > 0.50); CREATE DOMAIN Koruny NUMERIC(12,2);
- Jak lze zajistit bezpečnost a konzistenci dat v databázi? (Jmenujte jak možnosti v SQL tak i jiné.)**  
pomocí DCL: systém uživatelů, skupin a privilegií; pomocí transakcí: Trans. Control Commands – COMMIT, ROLLBACK, SET TRANSACTION, SAVEPOINT; pomocí služeb SRBD: údržba, záloha, obnova po pádu; pomocí HW: záložní zdroje, RAID disky;
- Jaké jsou nevýhody relačního modelu?**

- 1) neschopnost modelovat komplex. dat. struktury; 2) neschopnost agregace a generalizace; 3) nepodpora hlediska času a verzí objektů; 4) impedanci nesoulad (impedance mismatch – uspořádání veškerých dat do tab.);
43. **Definujte typ na řádek tabulky ADRESY s atributy ulice, číslo a město**  
CREATE ROW TYPE typ\_adresa (ulice ADRESY.ulice%TYPE, město ADRESY.mesto%TYPE);
44. **Charakterizujte oba dva typy triggerů**  
příkazové – implicitní, spustí se pouze jednou za příkaz na tabulce, nad kterou je vytvořen; řádkové – zápisem FOR EACH ROW, spustí při každé akci nad tabulkou pro kterou je definován (i vícekrát za příkaz);
45. **Rozdíl mezi architekturou klient-server a distribuovanou DB, výhody a nevýhody**  
Klient-server – výhody: centralizovaný, menší HW nároky na klienta, jednodušší na vývoj; nevýhody: delší odezvy od centrálního stroje;  
DDB – výhody: decentralizovaný, výkonnější, spolehlivější, lepší rozšiřitelnost; nevýhody: větší zatížení sítě, složitost, obtížné zavedení (přechod);
46. **Jaké jsou v SQL možnosti pro vytváření stromů?**  
1) SQL: příkazem SELECT s klauzulemi START WITH a CONNECT BY; 2) ANSI-SQL-99: příkaz WITH  
stromy se vytváří pomocí pre-order procházení
47. **Jak lze ošetřit víceklauzule transakce v SQL 99 a popište možnosti při neúspěšné transakci**  
1) obecně: vypnutím automatického commitu po každé transakci a používání COMMIT; 2) jen SQL99 pomocí BEGIN ATOMIC ... END; při neúspěšné transakci se provede její rollback;
48. **Granularita triggerů (jakou známe), 2 typy:**  
Granularita BEFORE (zabrání nevhodnému zpracování) a AFTER (načítání dat pouze 1x);  
1. typ: Daty řízeny: spouští se pro každý řádek, zápis pomocí klauzule FOR EACH ROW;  
2. typ: Událostmi řízeny, spouští se pro každou tabulku pouze jednou;
49. **Nevýhody boolského modelu dokumentograf. db + uveďte možnost nápravy**  
Nesnadná formulace dotazů, nemožnost rozlišení důležitosti termů dotazu, nemožnost automatické modifikace dotazu na základě odpovědi, nepřesné výsledky, nelze řídit velikost výstupu; možnost nápravy: použít rozšířený Boolovský (zavedení vah), nebo vektorový model;
50. **Kolekce a hnízděné tabulky, pravidla pro použití metod kolekce**  
kolekce: např. nekonečná neuspořádaná: CREATE TYPE kolekce AS TABLE OF varchar(20); hnízděné tabulky: CREATE TABLE tab ( ... ) NESTED TABLE (tab\_nested) STORE AS typ;  
pravidla: metody kolekce jsou použitelné pouze PL/SQL (ne v SQL);
51. **Typy spojení tabulek + podmínky**  
INNER JOIN ON; LEFT OUTER JOIN ON; NATURAL INNER JOIN; NATURAL RIGHT OUTER JOIN; FULL OUTER JOIN USING;
52. **Hlavní nástroje objektového přístupu v db (identita, návrh, aplikace) + teoretické východisko a příklad**  
Hlavní nástroje: ODL, OQL, OID (reference), objekty, návrh pomocí metod, seznamů, množin, hnízděných tabulek;
53. **Typy indexů, kdy pomohou a kdy ne**  
B-tree indexy: hodnoty v listech, které jsou oboustranně linkované, použitelné pokud dotaz omezuje prvních k sloupců indexu; Bitmapové indexy: pro každou hodnotu sloupce/výrazu se vytváří bin. řetězec obsahující 1 pro řádky s danou hodnotou, vhodné pro sloupce s nízkou selektivitou  
Oracle vytváří automaticky unikátní (pro PK, kandidátní klíče). KDY ANQ: pro FK, pro často kladené dotazy, pro výsledky s nízkou selekcí KDY NE: nad všemi sloupci, pro dotazy se selekcí nad 5%;
54. **Příklad temporální projekce**  
Zanznamenávání platnosti dat: sloupce „platnost\_od, platnost\_do, platu, vzámestnaneč“; → Jaká je platová historie?
55. **Vztah událostí a času v temporálních db – datové modely a podpora snapshot**: nepodporuje nic; valid-time: čas platnosti; transaction-time: trans. čas; atemporální: čas platnosti ,transak. čas.; temporální: čas platnosti nebo transakční čas;
56. **Předzprac. textu v dokumentografických db systémech - co si pod tím představujeme, ne/výhody**  
předzprac. textu: vytvořit model s co nejvíce informacemi a s efektivním vyhledáváním; zahrnutí lingvistické analýzy (desambiguace, lemmatizace), obecně: Filtrace a integrace dat – výběr atributů vhodných pro analýzu, ošetření nebo vyloučení dat chybných, chybějících, redundandních, irelevantních, konstantních; sjednocení formátů, měrných jednotek; numerické zakódování některých dat, sjednocení kódování; kategorizace a dichotomizace; dat Transformace - standardizace atributů (odstranění závislosti reálných atributů na jednotkách měření), normalizace objektů (odstranění závislosti na velikosti objektu), hlavní komponenty; Odvozování - odvozené atributy • agregované údaje; výhody: efektivnější vyhledávání; nevýhody: náročné;
57. **Popište Rule-Based optimalizaci a jaké má nevýhody.**  
Odvozuje plán ze syntaxe příkazů a existence indexů.  
Cena přístupu k podmnožině řádek v tabulce v klesajícím pořadí: Full scan (procházení celé tabulky a ověřování podmínky, vhodné při velkém % vyhovujících); Index-range-scan (vyhledávání intervalu v indexu a v nich ověření podmínky); Unique-index-scan (vyhledávání jediné možné řádky podle unikátního indexu) RowID-scan (vyhledávání na základě známé hodnoty fyzického identifikátoru v db);  
Nevýhodné při existenci více neunikátních indexů na jedné tabulce, použitím určitého indexu je možné znemožnit optimalizátoru použití výrazu v dotazu.
58. **Popište Cost-Based optimalizaci.**
- Založena na statistikách (počet různých hodnot ve sloupci, počet řádek v tabulce), počítá cenu zdrojů provedení operace. Dokáže rozlišit plány i pro různé hodnoty konstant v dotazu.
59. **Jaké jsou rozdíly boolského a vektorového modelu dokumentografických systémů DBIS?**  
Booleovského model: je obsaženo N dokumentů v databázi a k nim M termů, které je popisují (dokument je reprezentován termy). Zpracováním dokumentů na vstupu vzniká posloupnost dvojic <dok\_id, term\_id> (obecně n-tic); → nevýhody; Vektorový model: dokumenty reprezentovány vektorem vah termů = < i i i i m d w w w, 1, 2, , , ... , kde  
i j w, je míra důležitosti j-tého termu pro identifikaci i-tého dokumentu. Podobnost mezi vektorem dotazu a vektorem dokumentu dán podobnostní funkcí. Dokumenty na výstupu řazeny sestupně dle podobnosti, lze omezit velikost výstupu (u Bool. ne) a specifikovat minimální nutnou podobnost.  
Počet vrácených dokumentů na dotaz uživatele (120) z nich počet opravdu relevantních (90) a celkový počet relevantních v celém systému (300);
60. **Spočítat přesnost P a úplnost R.**  
(ne)vybrané, (ne)relativní: VR, VN, NR, NN; úplnost: R=VR/(VR+NR); přesnost: P=VR/(VR+VN);
61. **Co je Embedded SQL a kdy je výhodné ho použít namísto klasického SQL?**  
Vložený SQL je SQL napsaný v jiném programovacím jazyce, zpravidla Pascal, PL/I, Fortran, C, Cobol Java. Princip vnořeného SQL spočívá v tom, že umožňuje do některého z vyšších programovacích jazyků vkládat příkazy SQL označené standardním prefixem. Ty jsou pak překládány prekompilátorem na volání funkcí knihovny.
62. **Jaké jsou typy JDBC ovladačů (popište)?**  
1) využívá ODBC (přes JDBC-ODBC bridge), obtížně konfigurovatelné; 2) komunikace s nativním ovladačem; 3) komunikace s centrálním serverem (Network server); síťovým protokolem; 4) založen čistě na jazyce JAVA, přímý přístup do databáze;
63. **Co je to data cartridge (popište k čemu je to dobré, případně uveďte příklad)?**  
uživatelské rozšíření databáze Oracle (rozšíření serveru – manipule s daty probíhají na něm). Může definovat nové datové typy, indexy. Hodí se ke zpracování a manipulaci s komplexnějšími daty. Př.: finanční a právní systémy, multimédia, datové řady, statistické výpočty, Oracle Text (pro vyhledávání ve slopcích)
64. **Jaké vlastnosti má ideální temporální databáze?**  
Minimální rozšíření existujícího datového modelu, souvislá prezentace chování měnicího se v čase, snadná implementace, vysoký výkon. Dosažení v realu je prakticky nemožné.
65. **Co je to aktivní databáze?**  
DB, v níž jsou za určitých podmínek automaticky spouštěny procedury (triggery) při provádění některého příkazu SQL a tím hlídána referenční integrita, aplikační pravidla, bezpečnost, vytvářena synchronizace tabulek, audit, statistiky, zálohování;
66. **Jaké výhody přináší jazyk SQL3 (SQL99)?**  
regulérní výrazy, rekurzivní dotazy, triggery, nescalární typy, objektově orientované záležitosti;
67. **Jaké typy temporálních dotazovacích jazyků znáte ?**  
relační: HQL, HSQL, TSQL, TSQL2 objektově orientované: OQL, OSQL
68. **Temporální databáze doplňte údaje:**  
modely času (lineární, větvený, cyklický); modely hustoty (diskrétní, hustý, spojitý);
69. **Jak zlepšuje propustnost komunikačního kanálu PL/SQL ?**
70. **Přidělování práv**  
GRANT OPTION: propůjčování práv dalším; SET ROLE: nastaví roli, ostatní se vypnou;
71. **Na co musíme dávat pozor při návrhu databáze?**  
Maximální důraz na přesunutí manipulaci dat na úroveň databáze (centrálně – integr. omez., triggery), protože se lépe ohlídá konzistence dat. Čištění nekonzistentních dat může být až nemožné;
72. **Popište mechanismus použití zpravy PREPARE TO COMMIT v distribuovanem DBS.**  
Použije se 2fázový potvrzovací protokol. 1) Pošle se na všechny uzly DDBS zpráva pro ukončení transakcí, které jsou nad daty připravenými ke commitu. 2) Po zprávě o potvrzení commitu všech uzlů, kterých se zpráva týká, se teprve přejde ke commitu na všech uzlech z připravených dat;
73. **Jak nastavit národní prostředí?**  
3úrovňové nastavení: 1) Pro klienta: přepínače – NLS\_ : lang, langure, territory, sort, dateformat, datelanguage, calendar; 2) Pro uživatele: proměnná prostředí; 3) Pro session: paramet pro konkrétní SQL dotaz;
74. **Vysvětlete rozdíl mezi multidatabázovými a federativními databázovými systémy.**
75. **Popište princip metody k-means (vzorec).**
76. **Popište na příkladu hierarchický pohled na databázi.**
77. **hierarchický = síťový model**  
kolekce podobných záznamů vytvářejí seznamové struktury se záhlavím; ze 60. let;
78. **Způsoby zprac. SQL dotazu, ne/výhody statického SQL**