Vypracované otázky **PB154 - Základy databázových systémů** (částečně recyklované) z fi.muny.cz.
Pokud se vám hodily už vypracované otázky a považujete za cenné, když jsou dostupné otázky z minulých let, pokuste se zapamatovat si ty své a následně je dát na fi.muny.cz. Budoucí generace to ocení. :-)

Popište, co znamená?
Cizí klíč: v prostředí relačních databází definuje vztah mezi dvěma tabulkami takový, že hodnota v určeném sloupci musí existovat v jiné (primární) tabulce. Tím je definováno integritní omezení, které do tabulky položky umožní vložit jen povolené hodnoty. Je tím vlastně vytvořeno spojení jednoho nebo více sloupců se sloupcem nebo více sloupci jiné („cizí“) tabulky. Tomu se též říká reference nebo odkaz.
Cizí klíč umožňuje definovat akce, které mají nastat při pokusu o změnu nebo mazání záznamů v cizí tabulce.

Superklíč: Určuje každý řádek tabulky jednoznačně, ale není nutně minimální.

Kandidátní klíč: v relačním modelování označuje sloupec nebo kombinaci sloupců, ve kterých mají všechny řádky tabulky své hodnoty unikátní. Každý kandidátní klíč tak umožňuje jednoznačně identifikovat každý řádek tabulky. Kandidátní klíč je nejmenší možná taková množina sloupců.

Primární klíč: je pole nebo kombinace polí (jeden vybraný z kandidátních), jednoznačně identifikující každý záznam v databázové tabulce. Žádné pole, které je součástí primárního klíče, nesmí obsahovat hodnotu NULL. Každá tabulka má mít definovaný právě jeden primární klíč.

Integritní omezení se mohou týkat jednotlivých hodnot vkládaných do polí databáze (například známka z předmětu musí být v rozsahu 1 až 5), či může jít o podmínku na kombinaci hodnot v některých polích jednoho záznamu (například datum narození nesmí být pozdější než datum úmrtí). Integritní omezení se může týkat i celé množiny záznamů daného typu – může jít o požadavek na unikátnost hodnot daného pole či kombinace polí v rámci celé množiny záznamů daného typu, které se v databázi vyskytují (například číslo průkazu v záznamech o osobách).

Velmi často používaným integritním omezením v relačních databázích je tzv. referenční integrita. Jedná se o požadavek, aby pro pole záznamu, jež má obsahovat odkaz na jiný záznam někde v databázi, takový odkazovaný záznam skutečně existoval.

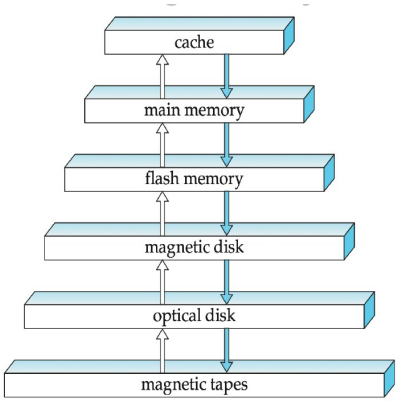
Trigger v databázi definuje činnosti, které se mají provést v případě definované události nad databázovou tabulkou. Definovanou událostí může být například vložení nebo smazání dat.

Běžný rotační pevný disk?
Popište 3 základní složky časových nákladů potřebných ke čtení jednoho bloku
Zařaďte disk do hierarchie pamětí.

Čtení bloku:
1. Nastavení hlav nad správnou stopu.
2. Vyčkání, až bude správný sektor pod hlavou.
3. Přečtení a odeslání informace.
4-11 milisekund

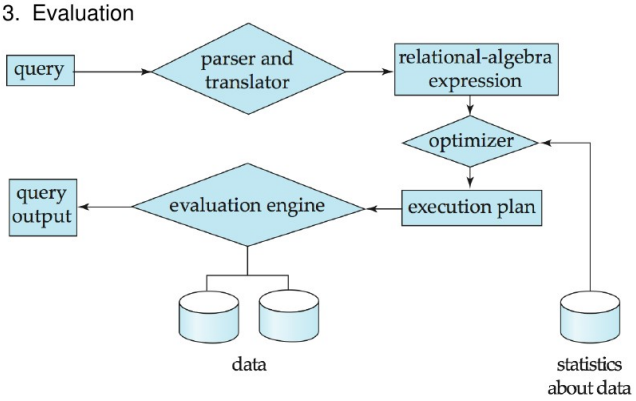
Klasifikace paměťových médií:
Rychlost přístupu
Cena za jednotku paměti, typicky jeden bit.
Spolehlivost: Ztráta dat při výpadku energie. Frekvence HW poruch.
Volatilní: Data jsou ztracena při odpojení napájení.
Nevolatilní: Data zůstávají I po odpojení napájení.

Hierarchie pamětí:



Nakreslete schéma kroků zpracování dotazu a popište?

1. Parsing and translation
2. Optimization
3. Evaluation



1. rozbor a překlad (parsing and translation)
2. překlad dotazu do vnitřní formy a následně relační algebry, kontrola syntaxe a verifikace relací
3. optimalizace (optimization) zkoušení nalézt nejlevnější plán (posloupnost kroků) pro vyhodnocení dotazu (výrazy relační algebry lze totiž vyjádřit více ekvivalentními výrazy a každý výraz relační algebry může být různě složitý na provedení).
4. vyhodnocení (evaluation) převzetí a provedení plánu a vrácení odpovědi na dotaz.

Pro optimalizaci se využívají známa statistická data o databázi (počty a velikosti sloupců a hodnot). Hodnocení je založeno na odhadu doby, kterou bude asi vyhodnocení trvat. Ta může být ovlivněna I rychlostí procesoru, disků,... počtem sektorů k přečtení a zapsání, počtem přístupů na disk,...

Popište, co je sekvenční soubor a uveďte příklad souboru s vyhledávacím klíčem UČO Zkonstruujte nad souborem index pro jméno a uveďte typ indexu?
Sekvenční soubor ukládá data uspořádaně za sebou s ohledem na hodnotu vyhledávacího klíče pro každý záznam. Používá se přetoková oblast pro vkládání záznamů, na které není místo. Musí se aktualizovat ukazatele a čas od času provést náročná reorganizace.
Sobor bude obsahovat řádky uspořádané podle UČO.

Hustý index – indexový záznam je uložený pouze pro každou hodnotu vyhledávacího klíče (ale stejné hodnoty klíče se v indexu neopakují)
(Každé různé jméno by mělo vlastní záznam v indexu a ukazatel na něj v indexované tabulce.)

Řídký index – ndexové záznamy jsou pouze pro některé hodnoty vyhledávacího klíče.
Pro nalezení záznamu s vyhledávacím klíčem K musíme:
Nalézt indexový záznam s největším vyhledávacím klíčem menším než K.
Prohledat sekvenčně soubor od tohoto záznamu

Méně prostoru pro uložení indexu a méně udržujících operací při vkládání a mazání záznamu.
Obecně je ale při vyhledávání pomalejší než hustý index.
Vhodné řešení je řídký indexový soubor pro každý blok v souboru.
(Vybrat jenom některé položky soubotu, tu mají záznam s ukazatelem v indexu.)

Jiné způsoby ukládání do souboru:
Halda: Sama se uspořádává.
Hash: Funkce počítá podle obsahu záznamu, kam má být uložen. Může to špatně dopadnout ve chvíli, kdy je mnoho záznamů mapovaných do stejného místa. Používají se přetokové oblasti, které se prohledávají sekvenčně. Hashovací funkce musí být dobře navrhnutá, aby rozmisťovala data rovnoměrně.

Co je pohled View v SQL a jakým způsobem je definován + příklad relace a pohledu Uveďte dva způsoby používání pohledů?
V některých případech není žádoucí, aby uživatel viděl všechna data z databáze. Pohledy slouží k zakrytí některých dat pro různé uživatele.
create view POHLED as < SQL výraz >
Pohled uloží výraz, který se následně substituuje za název relace.
create view faculty as select ID, name, dept_name from instructor

Co je slotted page, jakým způsobem řeší fragmentaci volného místa?
Způsob ukládání záznamů v souboru.
Hlavička souboru obsahuje údaj o počtu záznamů, ukazatel na začátek volného místa a ukazatele na jednotlivé záznamy. Záznamy mohou být v soboru přeskupovány tak, aby mezi nimi nevznikalo nevyužitelné prázdné místo.
Záznamy mohou být proměnné délky.

Co je RAID, výhody použití, popis RAIDů?
Využití mnoha disků, které pracují paralelně, čímž se zvedá rychlost a kapacita. Redundance dat vytváří odolnost vůči poruchám.
Ukládají se informace, ze kterých je možné případná poškozená data zrekonstruovat.

Mirroring (shadowing):
Každý logický disk se skládá z dvou disků. Data jsou kládána paralelně stejná na oba disky. Pokud jeden selže, data zůstávají na druhém. Ke ztrátě dat by došlo pouze pokud by selhaly oba disky nebo druhý před nahrazením prvního.

Bit-level striping:
Každý bit každého bajtu ukládaných dat se uloží na jiný disk. V poli osmi disků, by každý bajt byl po bitech uložený na všech osmi discích. Rychlejší čtení než z jednoho disku, ale horší doba vyhledání záznamu.
Není dnes příliš používáno.

Block-level striping:
Každý blok ukládáaného souboru se uloží na jiný disk. Dotazy na poskytnutí bloků mohou fuungovat paralelně, pokud jsou bloky na různých discích.

RAID 0:
Používá block striping, nemá redundantní data. Využití v aplikacích vyžadujících rychlost, kde nezáleží na ztrátě dat.

RAID 1:
Disky zrcadleny, využívá block striping. Velká rychlost zápisu. Použití například pro ukládání logů.

RAID 2:
Bit striping kombinovaný s automatickou opravou chyb pomocí opravujícího se kódování.

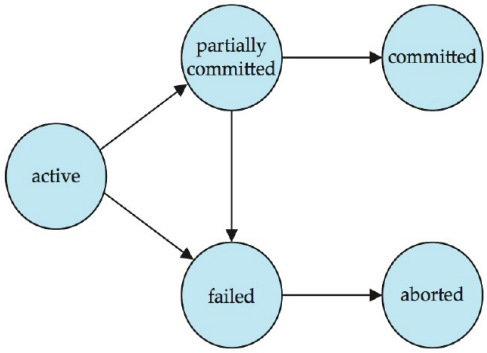
RAID 3:
Využívá disky pro ukládání paritních bitů, které slouží v kombinaci s ostatními disky k rekonstrukci dat. (XOR s daty na ostatních discích).

RAID 4:
Využívá block striping a paritní disk na ukládání celých paritních bloků. Protože paritní blok se ukládá pro každý zápis, paritní disk je slabé místo, které může zpomalovat.
RAID 5:
Paritní bloky se ukládají přímo na discích raidu na dalším disku v pořadí modulo počet disků + 1, ne na zvláštním disku mimo. Rychlejší než RAID 4 a odstraňuje problém s jedním vytíženým paritním diskem.

RAID 6:
Stejný princip jako RAID 5, ale navíc se ukládají redundantní data pro případ selhání.

RAID může být softwarový nebo hardwarový (složitější, více možností k poruše, používá nevolatilní RAM paměť)

Co je plán zpracování transakcí a co musí splňovat, co je sériový plán (příklad)?
Nakreslit stavový diagram?



Aktivní: Počáteční stav transakce
Částečně provedená: Po vykonání posledního příkazu transakce.
Provedená: Po úspěšném dokončení a zapsání.
Selhala: Při chybě provádění.
Zrušená: Ve chvíli, kdy transakce selhala a databáze se vrátila do původního stavu před provedením. Transakce může být znovu spuštěna nebo zahozena.

Transakce (transaction) je posloupnost jednotlivých příkazů, která má určitý význam
vlastnosti zarucujici integritu:
- atomicita: transakce musí být buď provedena celá nebo vůbec, nesmí dojít ke ztrátě dat při selhání systému v průběhu.
- konzistence: databáze musí být před prováděním transakce v konzistentním stavu a musí v něm být I po provedení. Například součty hodnot musí odpovídat před I po.
- izolovanost: každá transakce z více najednou prováděných transakcí nesmí vědět o jiných a její mezivýsledky musí být schovány před ostatními.
- trvanlivost: po úspěšném dokončení transakce, změny musí v databázi přetrvat (i v případě selhání systému).

T_1	T_2
	read (A)
	$temp := A * 0.1$
	$A := A - temp$
	write (A)
	read (B)
	$B := B + temp$
	write (B)
	commit
read (A)	
$A := A - 50$	
write (A)	
read (B)	
$B := B + 50$	
write (B)	
commit	

Serializace je způsob provádění transakcí jednu po druhé, což udržuje databázi v konzistentním stavu, protože transakce vždy přistupují k databázi po všech předchozích změnách. Serializovatelnost současně běžících transakcí znamená, že by mohly být provedeny seriově a tudíž neporuší konzistenci databáze.

Porovnejte, popište od nejlepší k nejhorší hashování, B strom a B+ strom?
Hledání prvků s konkrétní hodnotou: H, B, B+
Hledání prvků z intervalu: B+, B, H
Vkládání prvku: H, B+, B

Princip algoritmu pro spojování relací pomocí hashování na příkladu?
Hahovací funkce rozdělí tabulky do odpovídajících si bloků, které mají stejnou hodnotu hashovací funkce na spojovacích atributech. Bloky je potom potřeba porovnat a spojit jen odpovídající si mezi sebou, nikoli všechny se všemi. Vhodné pro natural join.

1.) Co jsou složky ER-modelu?
a) Entitni mnoziny
b) Mnoziny vztahu
c) Atributy

a) Databaze muze byt postavena na modelu mnoziny entit nebo vztahu mezi entitami.
ENTITA - je hmotny nebo abstrakni objekt, ktery existuje v realnem svete a je jednoznacne odlisitelny od ostatnich obektu (osoba nebo udalost).
ENTITNI MNOZINA - jemnozina entit stejneho typu, ktere sdileji stejen vlastnosti (mnozina studentu MU)

b) VZTAH - je spojitost mezi nekolika entitami (vkladatel: zakaznik-ucet)
MNOZINA ZVTAHU - je matematicka relace mezi minimalne dvema entotami, z nichcz kazda prochazi z urcite mnoziny entit
- muze mit take sve atributy (vkladatel: datum pristupu)
STUPEN MNOZINY VZTAHU - je pocet entitnich mnozin, ktere se sdileji na mnozine vztahu
- vetsina je binarni, pripadne ternarni

c) entita je predstavovana mnozinou atributu, vlastnosti, jimiz jsou obdareni vsichni clenove dane entitni mnoziny (student jmeno a cslo)
DOMENA - je mnozina vsech moznych hodnot atributu
TYPY - jednoduche a slozene(adresa), jednohodnotove (UCO), vicehodnotove {tel., te., ...}
nulove, odvozene (věk z data narození)

6.) Jakym zpusobem mohou diskova pole urychlit zpracovani pristupu k datum?
a) malych objektu
b) velkych objektu

a) Vesi pristupova rychlost
b) Velka propustnost dat

9.) Charakterizujte strucne jazyk pro definici dat (DDL) a jazyk pro manipulaci s daty(DML)?
DDL - jazyk pro definici databazoveho schematu umoznuje specifikaci
- schematu relace
- mnoziny atributu
- domenu hodnot spojenou s kazdym atributem
- intergritní omezení
- mnozinu indexu, ktere budou udrzovany pro kazdou relaci,
- bezpečnostní opatření
a informace o opravneni pristupu pro kazdou relaci, fyzickou strukturu ukladani na disk pro kazdou relaci.

DML - jazyk pro manipulaci s daty
Umožňuje pokládat dotazy do databáze a provádět modifikaci dat (insert, delete, update)
- NEPROCEDURALNI - uzivatel specifikuje, jaka data pozaduje
- PROCEDURALNI - uzivatel specifikuje navic i zpusob, jak uskutecnit pozadavek

10.) Definujte SELECT, PROJECT?
SELECT – odpovida operaci projekce v relacni algebre
- pouziva se k vypsani pozadovanuch atributu ve vysledku dotazu
- hvezdicka(*,asterisk) v klauzuli znaci vsechny atributy
- SQL umoznuje vyskyt duplicit
- klicove slovo DISTINCT uvedene bezprostredne za slovem SELECT zpusobi odstraneni duplicity
- klicove slovo ALL uvedene hned za slovem SELECT zpusobi ponechani duplicity
- muze obsahovat aritmeticke vyrazy(+,-,*,/) a operace na konstantach nebo atrinbutech n-tic

14.) Napis 3 metody implementace JOIN?
Hash-join, merge-join, nested-loop join, block nested-loop join, indexed nested-loop join

16.) Definujte fyzickou a datovou nezavislost?
Fyzicka uroven - popisuje zpusob ulozeni dat.
Logicka uroven - popisuje data ulozena v databazi a vztahy mezi nimi.
Nezavislost dat - modifikace schematu nebo prostredi jedne urovne nema vliv na schema vyssi urovne.

- rozhrani mezi ruznymi stupni a koponentami by mela byt dobre definovana, aby zmeny v nekterych castech nemely vyznamny vliv na jine casti
Fyzická datová nezávislost: aplikační programy jsou nezávislé na fyzické datové struktuře.
Logická datová nezávislost: aplikační programy jsou nezávislé na změnách v logické struktuře databázového souboru.

18.) Co je outer join a jake jsou jeho formy?
Rozšíření operace přirozené spojení, které zabraňuje ztrátě informací.
Spočítá operaci spojení a přidá n-tice z jedné relace, které neodpovídají n-ticím v druhé relaci k výsledkům operace spojení, řádky, které není s čím spojit jsou do výsledku zkopírovány a doplněny speciální hodnotou null, která vyjadřuje absenci hodnoty (hodnota je neznámá). Null hodnoty mají logickou hodnotu false.

20.) Kdy dojde k preteceni datove oblasti pri hashovani a jak tomu predejit?
Pokud hashovací funkce namapuje více záznamů do jednoho místa (více záznamů má stejný vyhledávací klíč, špatně navrhnutá funkce). Dojde k přetečení bucketu a je vytvořen další zřetězený. Přetokové oblasti se potom prohledávají sekvenčně.
Použití rozšiřitelného hashování, které zvětšuje nebo zmenšuje slovník podle potřeby.

23.) Rozdil mezi relacnim schematem a relaci?
Relace je množina uspořádaných n-tic a odpovídá tabulce.
Uspořádaná n-tice odpovídá řádku tabulky. Je prvkem relace.
Relační schéma je uspořádaná n-tice, nesmí být prázdné a odpovídá záhlaví tabulky.

25.) Rozdil mezi kartezskym soucinem a natural-join?
Kartézský součin spojí všechny řádky první tabulky se všemi řádky druhé tabulky, vzniknou všechny kombinace.
Natural join spojuje jen řádky tabulky, které mají stejnou hodnotu jednoho atributu.

32.) Co je to víceúrovňový index a kdy se pouziva?
Primární index je setříděný podle vyhledávacího klíče.
Sekundární index je setříděný jinak než sekvenční uspořádání souboru.

Víceúrovňový index: Pokud se index nevejde do paměti, je rozdělen řídkým indexem, který specifikuje kde hledat. Pokud se ani řídký index nevejde do paměti, je přidána další úroveň. Všechny indexy je potřeba updatovat v případě změny dat.

33.) Co je to datovy model a jake jsou jeho zakladni formy?
Soubor nastroju pro popis dat. Objektove logicky modey - OO model, funkcionalni model...
Zaznamove logicke modely - relacni, sitovy, hiearchicky