

Svar till tentamen 2002-12-17

DATABASTEKNIK - 1DL116

5 poäng

Datum	Tisdagen den 17:e December
Tid	14:00 - 19:00
Jourhavande lärare	Kjell Orsborn, tel. 471 1154 eller 070 - 425 06 91
Hjälpmedel	inga

Anvisningar:

Läs igenom hela skrivningen och notera eventuella oklarheter innan du börjar lösa uppgifterna.

Förutom anvisningarna på skrivningsomslaget gäller följande:

- För att undvika misstolkningar - skriv tydligt och förklara så klart som möjligt. Lösningar som inte kan läsas eller tolkas kan naturligtvis inte ge några poäng.
- Formulera er kortfattat utan onödigt långa förklaringar.
- Antaganden utöver de som står i uppgiften måste anges. Gjorda antaganden får förstås inte förändra den givna uppgiften.
- Skriv endast på en sida av papperet och använd ett nytt papper för varje uppgift för att underlätta rättning och minska risken för missförstånd.

För godkänt krävs cirka 50% av maxpoäng.

1. Datamodeller

4p

Förklara begreppen *primärnyckel* (eng. primary key) för relationsdatamodellen och *objektidentifierare* (eng. object identifier) i en objekt-datatmodell samt jämför deras viktigaste egenskaper.

Svar: En primärnyckel är en minimal supernyckel, utvald bland kandidatnycklarna att utgöra nyckel för en relation. En minimal supernyckel består av en minimal delmängd av relationens attribut som unikt identifierar alla tupler i relationen. 1p

En objektidentifierare är en unik och ofta logisk och systemgenererad identifierare som används för att unikt identifiera objekt under hela dess existens, samt för att hantera referenser mellan objekt. 1p

De huvudsakliga skillnaderna mellan en P.N. och en OID är att en P.N. unikt identifierar tupler (rader) i en tabell medan en OID unikt identifierar objekt i hela databasen. Vidare så är en OID alltid systemgenererad, ändrar aldrig värde och används internt för att referera till objekt. En P.N. utgörs normalt av en eller flera attribut (kolumner) från en relation, de väljs av databaskonstruktören och kan följaktligen ändras eller korrigeras. 2p

2. Konceptuell modellering:

4p

a) Vad är skillnaderna mellan Entitets-Relationsmodellering (eng. entity-relationship modeling) och Utökad ER-modellering (eng. enhanced/extended entity relationship modeling)?

Svar: Förutom begreppen entitetstyp, attribut och relation som är de huvudsakliga modelleringsbegreppen i ER-modellering så har Utökad ER-modellering kompletterats med begrepp för att representera arv av egenskaper genom supertyp/subtyp (superklass/subklass) relationer och unionstyper tillsammans med villkor för att representera begränsningar som partiell och total deltagande hos föregående begrepp samt överlappande (overlapping) eller åtskiljda (disjoint) hos specialiserings- eller generaliseringsrelationer. 2p

b) Hur representerar man begreppen *entitetstyp* (eng. entity type) och attribut (eng. attribute) från en E-R modell i relationsmodellen och i en objekt-orienterad datamodell?

Svar: I relationsmodellen representeras en entitetstyp som en relation (tabell) och entitetstypens attribut som attribut i relationen (en kolumn i tabellen). I en objekt-orienterad datamodell representeras en entitetstyp som en klass (eller typ) och attribut som attribut hos klassen. 2p

3. Fysisk datalagring - index:

6p

Beskriv indexstrukturen hos följande indexeringsmetoder:

a) Hash-index (för extern hashning)

Svar: Ett hashindex består av en datafil uppdelad i ett antal datablock eller hinkar (eng. buckets). Dataposter fördelas till datablocken och återsöks med hjälp av en hash-funktion som appliceras på hash-fältet i filen. Hash-funktionen returnerar därvid adressen till ett datablock och fördelar på så sätt dataposter, vid insättning, till olika datablock. På motsvarande sätt vid sökning av värden med avseende på hash-fältet så kommer hash-funktionen att ge adressen till det datablock (eller hink/bucket) där posten är lagrad. 2p

b) B⁺-träd

Svar: Ett B⁺-träd består av en balanserad trädstruktur med en ordnad mängd noder med sökvärden: enrot, inre noder och lövnoder. Ordningstalet p på trädet avgör storleken på trädet där inre noder har som minst p/2 och som mest p trädpekare (utom roten). Vidare har en inre nod med q st pekare (q < p) q-1 st sökvärden. Interna noder består av en ordnad mängd av pekare p och sökvärden k enligt <p₁, k₁, p₂, k₂, ..., p_{q-1}, k_{q-1}, p_q>, där k₁ < k₂ < ... < k_q. För alla sökvärden x i ett delträd pekat på från p_i gäller att k_{i-1} < x < k_i för 1 < i < q; x < k_i för i = 1; och k_{i-1} < x för i = q. Lövnoderna har strukturen <<pr₁, k₁>, <pr₂, k₂>, ..., <pr_{q-1}, k_{q-1}>, pr_{next}> där q < p, varje pr_i är en datapekare (eller datablock-pekare) och pr_{next} pekar vidare till nästa lövnod. Löven finns all på samma nivå och innehåller minst p/2 värden. (svaret kan ges en mycket mer kortfattat form mha figurer). 2p.

Samt diskutera:

c) För vilka typer av databassökningar kan man, och kan man inte, dra nytta av hashindex respektive B⁺-träd.

Svar: Hash-index stödjer mycket snabba sökningar av dataposter där sökvillkoret består av ett likhetsvillkor för ett enda datafält (indexeringsfältet). Hash-index kan däremot ej utnyttjas för att effektivisera sökningar där sökvillkoret inbegriper någon form av ordning av dataposterna. B⁺-träd är lämpliga för att exempelvis effektivisera ordnade sökningar där sökvillkoret exempelvis innehåller selektion av ett intervall av dataposter där intervallet begränsas av värden på indexeringsfältet. Sökning i frågor där sökvillkoret berör icke-indexeringsfält förbättras ej av exempelvis B⁺-träd-index (och inte heller någon annan typ av index). 2p

4. Relationsalgebra:

2p

Antag att vi har två stycken relationer (tabeller) med följande scheman (där xID bestämmer nycklar):

CIRKEL (CID, CNAMN, RADIE, PUNKTID)

PUNKT (PID, PNAMN, X-KOORDINAT, Y-KOORDINAT)

Uttryck i relationsalgebra följande fråga: Vilka nycklar, namn, radier och x-koordinater för deras centrum har de cirklar som har en radie under 50.0 (cm), och som har sin centrum-punkt i det positiva halvplanet $x > 0$?

Svar: `<cid,cnamn,radie,x-koordinat>`
`(radie < 50.0 AND x-koordinat > 0.0 (CIRKEL punktId = pid PUNKT))`

där `representerar` operatören naturlig join. 2p

5. Integritetsvillkor (eng. integrity constraints): **4p**

Förklara inom relationsdatamodellen följande begrepp:

a) entitetsintegritet (eng. entity integrity).

Svar: Entitetsvillkor uttrycker att ingen primärnyckel får anta värdet NULL så att alla tupler i en relation kan identifieras unikt. 2p

b) referensintegritet (eng. referential integrity).

Svar: Referensintegritet kräver att om en tupel i en relation refererar till en annan relation så måste den referera till en existerande tupel. 2p

6. Transaktionshantering: **4p**

Förklara, gärna med en bild (ledning: tillståndsdigram - eng. state transition diagram), de olika tillstånden som en transaktion genomgår från det att den startas tills dess den avslutas (skillnaden mellan att transaktionen genomförs tillfullo och att den avbryts skall framgå)

Svar: En transaktion går i ett *aktivt tillstånd* (eng. active state) direkt den startar där den kan utfärda läs- och skrivoperationer. När transaktionen avslutas övergår den i ett *partiellt överlämnat tillstånd* (eng. partially committed state). I detta tillstånd behöver man kontrollera att inget kan förhindra att transaktionen permanentas (vissa transaktionsprotokoll möjliggör detta genom att förändringar av databasen registreras i loggen). Om allt är OK så har transaktionen nått sin *överlämningspunkt* (eng. commit point) och kan övergå i ett *överlämnat tillstånd* (eng. committed state). En transaktion kan också övergå i ett *falerat tillstånd* (eng. failed state) från sitt partiella överlämnade tillstånd eller från sitt aktiva tillstånd om någon kontroll gått fel eller om transaktionen avbrutits. En falerad transaktion kan behöva rullas tillbaka för att eliminera dess effekt. Överlämnade och falerade transaktioner lämnar slutligen systemet genom att övergå i ett *avslutat tillstånd* (eng. terminated state). 4p

7. Distribuerade databassystem:

4p

Förklara de inom distribuerade databasområdet använda begreppen distributionstransparens (eng. distribution transparency), fragmenteringstransparens (eng. fragmentation transparency), samt replikeringsstransparens (eng. replication transparency).

Svar:

distributionstransparens eller *nätverkstransparens* innebär att användaren ej upplever närvaron av ett nätverk i ett distribuerat databassystem. Detta innefattar dels *lokaliseringstransparens* (eng. location transparency) som innebär att kommandon som används är oberoende av var data är lokaliserat och av lokaliseringen av systemet som utfärdat kommandot och dels *namngivningstransparens* (eng. naming transparency) som innebär att när ett objekt är namngivet så kan det därefter entydigt identifieras och accessas utan vidare specifikation. 2p

fragmenteringstransparens gör användaren omedveten om existensen av fragment, dvs uppdelning av en relation horisontellt eller vertikalt i delrelationer som är placerade distribuerat i systemet. 1p

replikeringsstransparens gör användaren omedveten om existensen av kopior av data, dvs kopior av data kan vara lagrade på flera noder i det distribuerade systemet (för att åstadkomma bättre tillgänglighet, prestanda och pålitlighet). 1p

8. Objektdatabassystem:

4p

Jämför en relationsdatabashanterares (eng. relational database management system) och en objektdatabashanterares (eng. object database management system) egenskaper vad det gäller hantering av utbyggbarhet (eng. extensibility) och uttrycksförmåga (eng. expressability).

Svar: .

Utbyggbarhet:

Relationsdatabashanterare (RDBH) utbyggbarhet begränsas normalt till möjligheten att definiera lagrade procedurer (stored procedures) för att specificera ny funktionalitet. Data-modellen kan normalt inte byggas ut utan man är begränsad till datatyper och modelleringsmöjligheter som relationsmodellen tillhandahåller. Moderna databashanterare som förutom relationsmodellen inkluderar objekt-orienterade utbyggnadsmöjligheter klassificeras normalt som objekt-relationella databashanterare.

Objektdatabashanterare (ODBH), inkluderar både objekt-orienterade databashanterare (OODBH) och objekt-relationella databashanterare (ORDBH). Båda dessa kategorier av DBH stödjer utbyggnad av DBH med egna sammansatta datatyper och operationer. Dessutom kan man definiera sina egna datarepresentationer. Vidare för att fullständigt hantera

utbyggnad av en DBH behövs ytterligare utbyggnadsmöjligheter som endast stöds av vissa ORDBH. För att fullständigt integrera nya datastrukturer och operationer med en DBH kräver att man kan definera egna index, kostnader, och ev. kostnadsmodeller för att möjliggöra optimering av frågor som innefattar dessa nya konstruktioner. För att fullständigt hantera sådana utbyggnader krävs också att själva frågeprocesseringen kan byggas ut med exempelvis nya optimeringsalgoritmer.

Uttrycksförmåga: Man kan förstås hävda att uttrycksförmågan är lika stor hos RDBH och ODBH då man har tillgång till programmeringsspråk hos båda. Det anses dock allmänt att en objekt-orienterad datamodell (OODM) har en större uttrycksförmåga än en relationsdatamodell då en OODM tillhandahåller stöd för egendefinierade komplexa objekt som exempelvis strukturerade objekt sammansatta av atomiska datatyper och kollektionsdatatyper. I en RDBH är man däremot begränsad av den inbyggda uttrycksmöjligheterna i relationsdatamodellen där attributvärden hos relationer måste vara atomiska och enkelvärda.

4p