**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI**

**PILANI CAMPUS**

**SECOND SEMESTER 2015 – 2016**

**DATABASE SYSTEMS (CS F212/ IS F243)**

**COMPREHENSIVE EXAMINATION (PART-A)**

**Date: 07.05.2016 Weightage: 25 % (50 M)**

**Duration: 2 hours Type: Closed Book**

**Note: Answer all parts of the question together.**

**Answers must be brief.** Total no. of pages: 4

**Q1. Answer the following, (there is a single answer for each question): [10\*1=10 Marks]**

1. Relation C is the join of relation A and relation B on condition p. Which of the following statements must be true in all cases?
2. The cardinality of C is greater than the cardinality of A
3. The arity of C is greater than the arity of A
4. The cardinality of C is less than the cardinality of A
5. The arity of C is less than the arity of A
6. None of the above
7. By which term is deferred update also known?
8. NO-UNDO/NO-REDO
9. UNDO/NO-REDO
10. NO-UNDO/REDO
11. UNDO/REDO
12. Immediate update.
13. A relational table that represents a many-to-many relationship must have
14. A single attribute primary key
15. A composite primary key
16. A composite foreign key
17. None of the above
18. Which one of the following is Deadlock prevention algorithm?
19. wound-wait
20. wound-die
21. wait-wound
22. wait-commit
23. The solution to the problem of overflows in hash join is to adopt overflow resolution methods. Which of the below is the correct choice?

A. Partition into smaller partitions with a different hash function

B. Increase the no. of partitions by a fudge factor

C. Combine partitions recursively so that each one fits into memory.

D. None of these

1. In query processing, for an equality comparison on a non-key attribute with a primary index, the cost of the operation is?
   1. hi \* (tT + tS) + tS + b\* tT
   2. tS  + (b \* tT )
   3. (hi + n) \* (tT + tS)
   4. (log2 b) \* (tT + tS)
2. Which of the following is true for the selection operation to distribute over the theta join operation when all the attributes in θ1involves only the attributes of *E*1 being joined and θ2 involves attributes of both E1 and E2?
3. σθ1(E1  θ E2) = (σθ1(E1)) θ E2
4. σθ1∧θ2 (E1 θ E2) = (σθ1(E1)) θ (σθ2 (E2))
5. σθ1(E1  θ E2) = E1 θ (σθ1(E2))
6. σθ1(E1  θ E2) = (σθ1(E1)) θ (σθ1(E2))
7. RAID level 4 refers to which of the following?
8. Disk mirroring with block striping
9. Bit-interleaved parity organization
10. Block-interleaved parity organization
11. Block-interleaved distributed parity organization
12. Suppose that there is a multiple granularity tree representing the hierarchy of data items in the order A -> B-> C -> D. Suppose that a transaction T1 is holding a shared mode lock on D and another transaction T2 requests for a lock on data item C in exclusive mode, which of the following is true?
13. Transaction T2 succeeds in getting the lock on C
14. Transaction T2 has to wait to get the lock on C
15. Transaction T1 is aborted
16. Transaction T1 and T2 are in deadlocked state.
17. Which of the following protocols ensures recoverable schedules?
    * 1. 2-phase locking protocol
      2. Graph-based protocol
      3. Timestamp ordering protocol
      4. Validation-based protocol

**Q2. Answer the following: [5 + 2 + 3 = 10 marks]**

(a) Match the following terms on the left related to normalization of relations with their correct definitions on the right:

|  |  |  |  |
| --- | --- | --- | --- |
| A | Functional Dependency | 1 | The value of one attribute depends on a primary key of another relation |
| B | Referential Integrity | 2 | Contains no partial Dependencies |
| C | Transitive Dependency | 3 | Constraint between 2 attributes |
| D | 2NF | 4 | An Attribute that can be broken down into component parts |
| E | Composite Key | 5 | Functional Dependency Between non-key attributes |

(b) Consider a relational schema EMPDATA (emp\_code, city, state, pin\_code). For any emp\_code, there is only one city and state. Also, for any given city and state, there is just one pin\_code. In normalization terms EMPDATA is a relation in which highest normal form?

(c) Relation R has eight attributes ABCDEFGH with the set of functional dependencies as F={CH→G, A→BC, B→CFH, E→A, F→EG} so that F + is exactly the set of FDs that hold for R. How many candidate keys does the relation R have?

**Q3. Answer the following: [2 + 3 + 4 +4 = 13 marks]**

(a) Suppose that Transaction 1 transfers $100 from account A to B and Transaction 2 transfers 30% of balance from B to A. Explain the ACID properties of a transaction using the above example.

(b) Write the steps of the modified Time-stamping protocol with Thomas’ Write rule. Given a schedule with sequence of instructions as T1: read(Q), T2: write(Q), T1: write(Q), T3: write(Q). Is this schedule valid under timestamp ordering protocol. If not, suggest appropriate modifications to the protocol for making the above schedule possible?

(c) Given the following schedules S1 and S2, state whether each of S1 and S2 are conflict or view serializable? Justify each.

|  |  |
| --- | --- |
| S1 | S2 |
| |  |  | | --- | --- | | T1 | T2 | | Read(A) |  | | A=A-10 |  | |  | Read(A) | |  | Temp=2\*A | |  | Write(A) | |  | Read(B) | | Write(A) |  | | Read(B) |  | | B=B+10 |  | | Write(B) |  | | |  |  |  | | --- | --- | --- | | T1 | T2 | T3 | | Read(X) |  |  | |  | Read(Z) |  | | Read(Z) |  |  | |  |  | Read(X) | |  |  | Read(Y) | | Write(X) |  |  | |  | Read(Y) |  | |  | Write(Z) |  | |  | Write(Y) |  | |

(d) Trace the steps of the recovery algorithm on the following log. Identify the redo and undo transactions and the final values of all data items in the log. Assume that the system failure occurs at point A.

|  |  |
| --- | --- |
| <*T*0start>  <*T*0, *A*, 0, 10>  <*T*0 commit>  < *T*1 start>  <*T*1, B, 10, 20>  <*T*1 commit>  <checkpoint {*T*0, *T*1}>  < *T*2 start>  <*T*2, C, 10, 20>  <*T*2 commit>  <*T*3 start>  <*T*3, D, 0, 10>  <*T*4 start>  <*T*4, E, 0, 10> | /\* [CONTINUED..]  <*T*4, E, 10, 20>  <checkpoint {*T*3, *T*4}>  <*T*5 start>  <*T*5, *E*, 0, 30>  <*T*5, E, 30, 40>  <*T*5 commit>  <*T*6 start>  <*T*6, *F*, 0, 10>  <*T*6, F, 10, 20>  <*T*6 commit>  <*T*7 start>  <*T*7, *D*, 10, 20>  A: /\*System Failure\*/ |

**Q4.** Consider 2 relations, Manager(manager\_ID, name, address, project\_ID) and Project(project\_ID, area, cost) of which a natural join has to be computed. The memory size is 4 blocks and the no. of blocks in the outer relation “Manager” are 900 and that in the inner relation “Project” are 600. The no. of buffer blocks allocated to each relation is 1. Use merge join to formulate the cost expression for the no. of block transfers and disk seeks required for the above operation. Use the obtained expression to calculate the total cost if the following conditions hold:

**[1 + 4 = 5 marks]**

* + 1. Relations are sorted on “project\_ID”.
    2. Relations are unsorted

**Q5.** Consider a relational schema of shipping system where Sailor (sailor\_id, sailor\_name, rating, age) has 500 tuples, Boat (boat\_id, boat\_name, color) has 160 tuples and Reserves (sailor\_id, boat\_id, day, reservation\_name) has 1000 tuples. Given that the no. of red boats are 80 and no. of sailors with age less than 30 years are 200, we wish to find the id, name and age of sailors as per the following SQL query as given below: [**1 + 4 = 5 marks]**

SELECT\_S.sailor\_id, S.sailor\_name, S.age FROM Sailor S, Boat B, Reserves R WHERE  B.boat\_id = R.sailor\_id AND B.boat\_id = R.boat\_id AND B.color = “Red” AND S.age < 30;

* + 1. Convert the above SQL query into relational algebra expression and generate the expression tree.
    2. Use equivalence rules on this expression tree to find the equivalent least cost expression tree after query optimization.

**Q6**. Consider the below B+ tree: **[1+1.5 \* 4 = 7 marks]**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 9 |  | 13 |  | 16 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1 |  | 4 |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 9 |  |  | 10 |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 16 |  | 20 |  | 25 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 13 |  | 15 |  |

Perform the following operations on the above tree in the order as given below where the upper limit to the degree (no. of children or pointers) of the tree is 4 (i.e. no. of search key values in a node that can be allowed <= 3):

* + 1. Insert 11
    2. Insert 12
    3. Delete 13
    4. Delete 15
    5. Delete 1

**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI**

**PILANI CAMPUS**

**SECOND SEMESTER 2015 – 2016**

**DATABASE SYSTEMS (CS F212/ IS F243)**

**COMPREHENSIVE EXAMINATION (PART-B)**

**Date: 07.05.2016 Weightage: 15 % (30 M)**

**Duration: 1 hour Type: Open Book**

**Note: Answer all parts of the question together.**

**Answers must be brief.** Total no. of pages: 5

|  |  |
| --- | --- |
| **NAME:** | **ID:** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Q1** | **Q2** | **Q3** | **Q4** | **Total marks** |
|  |  |  |  |  |
| **Recheck requests (if any):** | | | | |

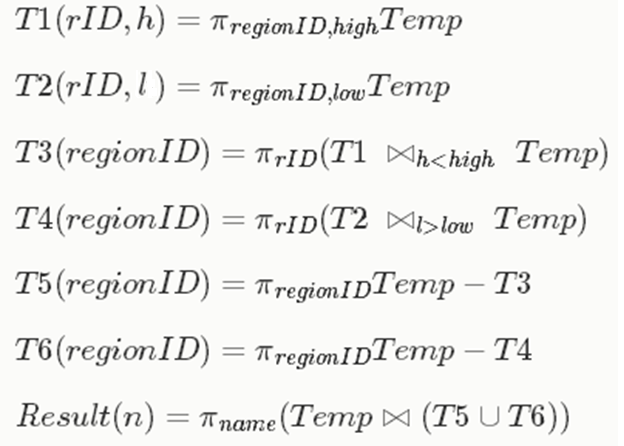
**Q1.**  Consider a database with the following schema: **[1 + 2 + 2 + 2 + 3 = 10 marks]**

|  |  |
| --- | --- |
| *Person* ( name, age, gender ) |  |
| *Frequents* ( name, pizzeria ) |  |
| *Eats* ( name, pizza ) |  |
| *Serves* ( pizzeria, pizza, price ) |  |

Write relational algebra expressions for the following queries:

1. Find the names of all females who eat either mushroom or pepperoni pizza (or both).
2. Find all pizzerias that serve at least one pizza that Amy eats for less than $10.00.
3. Find all pizzerias that are frequented by only females or only males.
4. Find the names of all people who frequent every pizzeria serving at least one pizza they eat.
5. Find the pizzeria serving the cheapest pepperoni pizza. In the case of ties, return all of the cheapest-pepperoni pizzerias.

**Q2.** Consider a relation *Temp* (regionID, name, high, low) that records historical high and low temperatures for various regions. Regions have names, but they are identified by regionID, which is a key. Consider the following query, give in words the result of this query? Justify at each step by giving proper explanation. **[5 marks]**



**Q3. Let E1 and E2 be two entities in an E/R diagram with simple single-valued attributes. R1 and R2 are two relationships between E1 and E2, where R1 is one-to-many and R2 is many-to-many. R1 and R2 do not have any attributes of their own. What is the minimum number of tables required to represent this situation in the relational model? Justify in steps using an appropriate example depicting the relational schema and the tables used.**

**[5 marks]**

**Q4.** Suppose that we need to create a dynamic hash structure on relational schema “Customer” with search key as the attribute “customer\_name” and the following tuples: **[10 marks]**

|  |  |  |  |
| --- | --- | --- | --- |
| Customer\_ID | Customer\_name | Phone\_no. | Email |
| 256 | John | 9998102000 | [john@gmail.com](mailto:john@gmail.com) |
| 312 | John | 8760981432 | john.d@gmail.com |
| 123 | Anna | 1222343241 | anna\_hilton@yahoo.com |
| 111 | Anna | 9098745987 | anna.123@yahoo.com |
| 432 | Anna | 0918228200 | [anna@gmail.com](mailto:anna@gmail.com) |
| 567 | Williams | 9897197265 | w[illiams.john@yahoo.co.in](mailto:illiams.john@yahoo.co.in) |
| 887 | Brown | 1782653092 | [brown\_1987@hotmail.com](mailto:brown_1987@hotmail.com) |
| 119 | Dan | 1615718998 | dan@hotmail.com |
| 876 | Suzan | 8759342987 | suzi@gmail.com |
| 765 | Suzan | 7863309000 | suzan@gmail.com |
| 234 | Mary | 2546338272 | [mary.g@yahoo.com](mailto:mary.g@yahoo.com) |
| 878 | Suzan | 7363864487 | suzi\_sim@hotmail.com |
| 871 | Suzan | 9836386447 | suzi\_moi@gmail.com |

The 32 bit hash values on “customer\_id” are h(customer\_id):

|  |  |
| --- | --- |
| Customer\_name | h(customer\_name) |
| John | 0011 1001 1000 0001 0110 1110 1111 0001 |
| Anna | 1011 0110 1110 0001 1010 0101 1101 0101 |
| Williams | 1100 0001 1100 1101 0001 1010 0101 0001 |
| Brown | 0011 1110 0001 1010 0001 0101 1110 0011 |
| Dan | 0111 1001 1111 0000 0101 1010 0011 1001 |
| Suzan | 0101 0001 1111 0101 1010 0010 1001 1111 |
| Mary | 0110 0001 1111 1001 0101 0001 1000 0011 |

Assuming the maximum bucket size is 3 tuples, clearly draw the step by step insertion of the tuples of the Customer relation in the dynamic hash table.