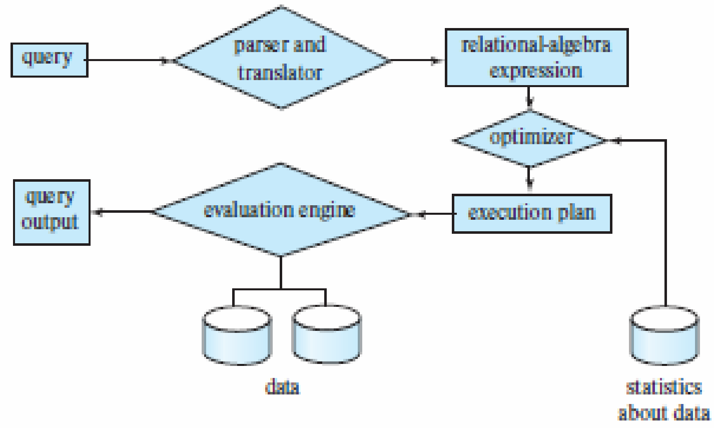
## 22CST41 - DATABASE MANAGEMENT SYSTEMS

## CONTINUOUS ASSESSMENT TEST 2 ANSWER KEY

| | | |
|---|---|---|
| | **PART A** | |
| 1. | Show the diagrammatic representation of query processing.<br><br> query<br>parser and translator<br>relational-algebra expression<br>optimizer<br>query output<br>evaluation engine<br>execution plan<br>data<br>statistics about data | 2 |
| 2. | Write the syntax to create index.<br>create index <index-name> on <relation-name> (<attribute-list>);<br>Example : create index dept index on instructor (dept name); | 1<br>1 |
| 3. | Define Multi-value dependency with example.<br>• In A->B, if A has single value, then B has multiple value<br>• Table should have atleast 3 column for it to have a multivalue dependency<br>• Relation(A,B,C) if it has multivalue dependency between A and B, then B and C should be independent of each other | 2 |
| 4. | Let R (A, B, C, D) be a relational schema with the following functional dependencies:<br>A → B, B → C, C → D and D → B. The decomposition of R into (A, B), (B, C), (B, D) check whether the decomposition is dependency preserved lossless decomposition? Justify.<br>The decomposition is dependency preserved lossless decomposition, since the intersection of all three relation is B and B is candidate key for R2. | 2 |
| 5. | Decompose the following empolyee into both lossy and lossless.<br>Employee (Employee_Id, Ename, Salary, Department_Id, Dname)<br>**Lossy**:<br>Employee (Employee_Id, Ename, Salary)<br>Department (Department_Id, Dname)<br>**Lossless**:<br>Employee (Employee_Id, Ename, Salary)<br>Department (Employee_Id, Department_Id, Dname) | 1<br>1 |
| 6. | Consider the relation scheme R = {E, F, G, H, I, J, K, L, M, N} and the set of functional dependencies {{E, F} -> {G}, {F} -> {I, J}, {E, H} -> {K, L}, K -> {M}, L -> {N} on R. What is the key for R?<br>**Key { E, F, H }** | 2 |
| 7. | What is extraneous attribute? Consider the following dependency and check whether extraneous attribute is present or not. If present remove the extraneous attribute. {AB → C, B → F, A → C, F →GH}<br>**B is extraneous attribute, Since A itself can determine C[A → C]** | 2 |
| 8. | Write the augmentation and pseudo transitivity rule.<br>**Augmentation rule :**<br>If α → β holds and γ is a set of attributes, then γα → γβ holds.<br>**Pseudotransitivity rule :**<br>If α → β holds and γβ → δ holds, then αγ → δ holds. | 1<br>1 |
| 9. | What is tertiary storage?<br>• The lowest level in the Storage device hierarchy.<br>• Referred as offline storage. | 2 |

|  |  |  |
|---|---|---|
|  | • External from computer system.<br>• Slowest speed and capable of storing large amount of data.<br>• Example, magnetic tape and optical disk, jukeboxes |  |
| 10. | Define Data dictionary.<br>Relational schemas and other metadata about relations are stored in a structure called the data dictionary | 2 |
|  | **PART B** |  |
| 11. | Consider the relation<br>Employees(name, emp_id, first_name, last_name, job, manager, hire_date, salary, commission, dept_id)<br><br>Write the SQL query for the following<br>i) Find out the names of all employees who belong to the same department as the employee 'William Smith' who is in department 100 and has an employee ID 40.<br>**Ans**:<br>**Select name from employees where dept_id in (select dept_id from employees where name='William Smith' and emp_id=40)**<br><br>ii) Find out the employees which belong to the department of 'William Smith' and have salary greater than the salary of 'Jessica Butcher' who has an employee ID of 40.<br>**Ans**:<br>**Select * from employees where dept_id in (select dept_id from employees where name='William Smith' and salary>( Select salary from employees where name ='Jessica Butcher' and emp_id=40))**<br><br>iii) Find out which of the employees have a salary less than that of the salary for the job ID 'FIN_ACT'.<br>**Ans**:<br>**Select * from employees where salary<(select min(salary) from employees where job ID='FIN_ACT')**<br><br>iv) Find the salaries for all employees who are not in the department 100<br>**Ans**:<br>**Select salary from employees where dept_id!=100** | 2.5<br>2.5<br>2.5<br>2.5 |
| 12. | Let R=(A, B, C, D, E, F, G, H, I, J) be a relational schema in which the following FDs are hold.<br>F={AB → C, A → DE, B → F, F →GH, D → IJ }. Decompose the above relation upto 3NF.<br><br>1NF : Already in 1NF<br>Candidate key {AB}<br>2NF :<br>Definition: should be in 1NF and no partial dependency<br>Decomposition : R1= {A,D,E,I,J}, R2={B,F,G,H}, R3={A,B,C}<br>3NF :<br>Definition : should be in 2NF and no transitive<br>Decomposition :<br>R1={A,D,E}<br>R2={F,G,H}<br>R3={A,B,C}<br>R4={D,I,J}<br>R5={B,F} | 2<br>2<br>3<br>3 |
| 13. | Describe the various RAID levels with neat sketch. Explain the factors to be considered into account in choosing RAID level.<br>Redundant Arrays of Independent Disks (RAID)<br>▪ **RAID: Redundant Arrays of Independent Disks**<br>• disk organization techniques that manage a large number of disks, providing a view of a single disk of<br>▪ **high capacity** and **high speed** by using multiple disks in parallel, | 7 |

| | | |
|---|---|---|
| | ▪ **high reliability** by storing data redundantly, so that data can be recovered even if a disk fails<br>▪ The chance that some disk out of a set of *N* disks will fail is much higher than the chance that a specific single disk will fail.<br>• E.g., a system with 100 disks, each with MTTF of 100,000 hours (approx. 11 years), will have a system MTTF of 1000 hours (approx. 41 days)<br>• Techniques for using redundancy to avoid data loss are critical with large numbers of disks<br>▪ **RAID Level 0**: Block striping; non-redundant.<br>• Used in high-performance applications where data loss is not critical.<br>▪ **RAID Level 1**: Mirrored disks with block striping<br>• Offers best write performance.<br>• Popular for applications such as storing log files in a database system.<br>▪ **RAID Level 5:** Block-Interleaved Distributed Parity; partitions data and parity among all *N* + 1 disks, rather than storing data in *N* disks and parity in 1 disk.<br>• E.g., with 5 disks, parity block for *n*th set of blocks is stored on disk (*n mod* 5) + 1, with the data blocks stored on the other 4 disks.<br>▪ **RAID Level 6**: P+Q Redundancy scheme; similar to Level 5, but stores two error correction blocks (P, Q) instead of single parity block to guard against multiple disk failures.<br>• Better reliability than Level 5 at a higher cost<br>▪ Becoming more important as storage sizes increase<br><br>**Factors in choosing RAID level**<br>• Monetary cost<br>• Performance: Number of I/O operations per second, and bandwidth during normal operation<br>• Performance during failure<br>• Performance during rebuild of failed disk<br>▪ Including time taken to rebuild failed disk | 3 |
| 14. | Elucidate the different types of organization of records in files. | |
| | **• Heap file organization:**<br>Any record can be placed anywhere in the file where there is space for the record. There is no ordering of records. Typically, there is either a single file or a set of files for each relation. | 2 |
| | **• Sequential file organization:**<br>Records are stored in sequential order, according to the value of a "search key" of each record. | 2 |
| | **• Multitable clustering file organization:**<br>Generally, a separate file or set of files is used to store the records of each relation. However, in a multitable clustering file organization, records of several different relations are stored in the same file, and in fact in the same block within a file, to reduce the cost of certain join operations. | 2 |
| | **• B+-tree file organization :**<br>The B+-tree file organization is related to the B+-tree index structure. It can provide efficient ordered access to records even if there are a large number of insert, delete, or update operations. Further, it supports very efficient access to specific records, based on the search key. | 2 |
| | **• Hashing file organization:**<br>A hash function is computed on some attribute of each record. The result of the hash function specifies in which block of the file the record should be placed. | 2 |