

**DEPARTMENT: COMPUTER SCIENCE**

|  |  |  |  |
| --- | --- | --- | --- |
| *Internal Assessment* | **Retest** | *Academic Year/Semester* | **2022-23 /IV** |
| *Subject* | **CST204-Database Management**  **Systems** | *Branch* | CSE/AI |
| *Date of Exam* | **27/06/2023** | *Duration* | **120 Min** |
| *Starting time* |  | *Max. Marks* | **60** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PART-A (*Answer all questions, each carries 3 marks*) Max Marks: 12** | | | | |
| ***Q.No*** |  | ***Marks*** | ***CO*** | ***Level*** |
| **1** | Define Participation constraint. Explain with neat diagrams Explain with suitable diagrams. | 3 |  |  |
| **2** | What is meant by Referential Integrity?. How it is implemented  using foreign key?. illustrate with example? | 3 |  |  |
| **3** | Demonstrate the working of HAVING clause in SQL | 3 |  |  |
| **4** | Define functional dependency, illustrate it with a relevant example, and explain the purpose or significance of functional dependency in database design? | 3 |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **PART-B (*Each question carries 12 marks)* Max Marks: 48** | | | | | | | |
| **5** | | a | | What are different types of DBMS architecture, Explain each one with neat diagrams. | 4 |  |  |
| b | | A university registrar’s office maintains data about the following entities: (a) courses, including number, title, credits, syllabus, and prerequisites; (b) course offerings, including course number, year, semester, section number, instructor(s), timings, and classroom; (c) students, including student-id, name, and program; and (d) instructors, including identification number, name, department, and title. Further, the enrollment of students in courses and grades  awarded to students in each course they are enrolled for must be appropriately modeled. Construct an E-R diagram for the registrar’s office. Document all assumptions that you make about the mapping constraints. | 8 |  |  |
| **OR** | | | | | | | |
| **6** | | a | | With neat diagram explain the 3 schema architecture | 4 |  |  |
| b | | Design an ER diagram for the given scenario; Suppose that you are designing a schema to record information about reality shows on TV.Your database needs to record the following information: \_ For each reality show, its name, genre, basic\_info and participants name. Any reality show has at least two or more participants. - For each producer, the company name, company country. A show is  produced by exactly one producer. And one producer produces exactly one show. - For each television, its name, start year, head office. A television may broadcasts multiple shows. Each show is broadcasted by exactly one television. -For each user, his/her username, password, and age. A user may rate multiple shows, and a show may be rated by multiple users. Each rating has a score of 0 to 10 | 8 |  |  |
|  | | | | | | | |
| **7** | | a | | Write Relational Algebra expressions for following queries.  ALBUMS(AlbumNo, AlbumName, ProducedBy, Year)  SONGS(SonsNo, SongsStart, Duration, AlbumNo)  SUNGBY(ArtistId, SongNo)  ARTIST(ArtistId, ArtistName)  a. Find the duration of the song with SongNo = 123:  b. Find the names of all artists who have sung a song in the album named "Thriller"  c. Find the names of all artists who have not sung any song  d. Find the names of all songs sung by the artist "Beyonce":  e. Find the names of all albums that have at least one song with a duration greater than 5 minutes: | 6 |  |  |
| b | | Suppose that we have an ordered file with 400,000 records stored on a disk with block size 4,096 bytes. File records are of fixed size and are unspanned,with record length 200 bytes.  1. How many blocks are needed for the file?  2. Approximately, how many block accesses are required for a binary search in this file? On an average, how many block accesses are required for a linear search, if the file is nonordered Based  on question give an example to illustrate that indexing can improve the search time. | 6 |  |  |
| **OR** | | | | | | | |
| **8** | | a | | Give the structure of a B+ tree data structure. What is its significance in database indexing over a B tree. | 6 |  |  |
| b | | Explain all Single level ordered indexes and Multilevel index with suitable diagrams | 6 |  |  |
|  | | | | | | | |
| **9** | | a | | STUDENT (enrol-no, s-name, phone, gender, DOB)  BRANCH (branch-name, branch-code, HOD-name)  COURSE (Subject-code, Subject-name, details, Branch-code) RESULT (enrol-no, Subject-code, marks, res, rank)  Based on above schema, write SQL queries  a). Create tables with all required integrity constraints.  b) Find the name of the youngest student who failed in more than two subjects.  b. Find the students who secured Grade A in all the courses offered by a specific department.  c. Find the age of all students. | 12 |  |  |
| **OR** | | | | | | | |
| **10** | **a** | | Event (eventid, name, description,city)  Participant (playerid, name, eventid, gender, year)  Prizes (prizeid, prize-money, eventid, rank,year)  Winners (prizeid, playerid)  Based on above schema, write SQL queries  a) Create tables with all required Integrity constraints.  b) Retrieve the name of events where all prize winners are females.  c) Retrieve the names of the persons who have won the highest number of 1st, 2nd and 3rd prizes. | | 12 |  |  |
|  | | | | | | | |
| **11** | **a** | | What is Normalization ?. Differentiate 1NF, 2NF, 3NF and BCNF? | | 6 |  |  |
| **b** | | A relation R(ABCDE) , and let the FD {A→B, B→C, C→D, D→A) decomposed into R1(ABC) and R2(CDE), Check if dependency preservation is satisfied in this decomposition. | | 6 |  |  |
| **OR** | | | | | | | |
| **12** | **a** | | Given R(ABCD), F= {AB, BC, CA}. Identify all Candidate Keys. | | 6 |  |  |
| **b** | | Consider a relation R(PQSTU) with FD {P→S, Q→S, S→T, TU→S, SU→P}, is decomposed into R1(P,T), R2(PQ), R3(QU), R4(STU), R5(PU). Check if this decomposition is lossless ?. | | 6 |  |  |

|  |
| --- |
| **Course Outcomes (CO):** |
| **CO I :** |
| **CO II :** |
| **Bloom’s Taxonomy Level:** |
| L1: Remember, L2: Understand, L3:Apply L4:Analyze |