|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **National University of Computer and Emerging Sciences, Lahore Campus** | | | | |
| C:\Users\saif\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.Word\final design.jpg | **Course Name:** | **Database Systems** | **Course Code:** | **CS2005** |
| **Degree Program:** | **BS (CS, DS, SE)** | **Semester:** | **Fall 2023** |
| **Exam Duration:** | **3 Hours** | **Total Marks:** | **75** |
| **Paper Date:** | **Mon 18-Dec-2023** | **Weight** | **50%** |
| **Section:** | **ALL** | **Page(s):** | **12** |
| **Exam Type:** | **Final Exam - SOLUTION** | **Total Questions:** | **9** |
| **Instruction/Notes:** | Scratch sheet can be used for rough work however, all the questions and steps are to be shown on question paper. ***No extra/rough sheets should be submitted with question paper***.  You will not get any credit if you do not show proper working, reasoning and steps as asked in question statements. | | | |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CLO No. | *3* | | *6* | | | | | | | Total |
| Q. No. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Marks |  |  |  |  |  |  |  |  |  |  |

|  |
| --- |
| **Roll No:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Section:** \_\_\_\_\_\_\_\_\_\_ **Name:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

**Q1.** *(5 points)* Discuss the ACID properties of a database transaction.

**Ans: See transaction chapter for ACID properties.**

**Q2.** *(10 points)* Map the following ER/EER Diagram into a relational model and specify all the constraints including primary key, foreign key, not null, and unique.

MANUFACTURER

N

M

PART

Contains

ORDER

N

LOCAL

FOREIGN

Has

CUSTOMER

1

N

1

Made\_by

**Relational Model**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **MANUFACTURER**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | Name | Street | City | Cnic | Passportno | Mtype | | | |
| **PART**   |  |  |  | | --- | --- | --- | | Part\_num | Description | name | |  | **CUSTOMER**   |  |  | | --- | --- | | id | name | |
| **ORDER**   |  |  | | --- | --- | | order\_num | id | | **Contains**   |  |  |  | | --- | --- | --- | | order\_num | id | Part\_num | | |

**Consider the following movie database for the next two questions:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Movie**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | movieId | movieTitle | releaseDate | runningTime | productionCost | studioId | | 1 | Giant | 1956-10-10 | 202 | 12233 | 2 | | 2 | Jack Reacher | 2012-12-21 | 145 | 600000 | 2 | | 3 | MalcomX | 1992-11-18 | 202 | 199999 | 2 | | 4 | Pacific Rim | 2013-08-23 | 130 | 455633 | 2 | | 5 | Fury | 2013-08-23 | 140 | 4656565 | 1 | | 6 | Transformers | 2013-08-23 | 150 | 4656565 | 1 | | **Genre**   |  |  | | --- | --- | | genreId | genreName | | 1 | Action | | 2 | Horror | | 3 | Sci-Fi | | 4 | War | | 5 | History | | 6 | Western | |
| **Studio**   |  |  |  |  | | --- | --- | --- | --- | | studioId | studioName | address | networth | | 1 | Universal Studios | California | 560000 | | 2 | Paramount Pictures | California | 10000 | | 3 | Warner Bros | California | 140000 | | **MovieGenre**   |  |  | | --- | --- | | movieId | genreId | | 1 | 2 | | 2 | 1 | | 2 | 4 | | 3 | 3 | | 4 | 5 | | 5 | 4 | | 6 | 1 | | 6 | 4 | |

**Q3.** *(10 points)* Consider the database state given above and for each of the following queries. Give the output for the database state given above.

1. Result ← Studio.studioId, Studio.studioName (σ Movie.studioId IS NULL (Studio ⟕ Studio.studioId=Movie.studioId Movie))
2. SELECT G.genreName, AVG(M.runningTime) AS avgRunningTime  
   FROM Genre G JOIN MovieGenre MG ON G.genreId = MG.genreId JOIN Movie M ON MG.movieId = M.movieId  
   GROUP BY G.genreName

HAVING AVG(M.runningTime)>140;

**Ans: : a)**

|  |  |
| --- | --- |
| **studioId** | **studioName** |
| **3** | **Warner Bros** |

**b)**

|  |  |
| --- | --- |
| **genreName** | **avgRunningTime** |
| **Action** | **147** |
| **~~History~~** | **~~130~~** |
| **Horror** | **202** |
| **Sci-Fi** | **202** |
| **War** | **145** |

**Q4.** *(15 points)* Consider the above movie database for the following problems.

1. Write SQL and RA statement to list the movies titles that have genres ‘Action’ as well as ‘War’.
2. Write SQL and RA statement to identify studios with more than 2 genres associated with their movies.
3. Create a View that lists the total number of movies for each genreName.

SQL.

select Movie.movieTitle from Movie where Movie.movieId in(

select movieId from MovieGenre where genreId=(

select Genre.genreId from Genre where genreName='War')

intersect

select movieId from MovieGenre where genreId=(

select Genre.genreId from Genre where genreName='Action'))



SQL.

SELECT studioId, COUNT(DISTINCT genreId) AS genreCount

FROM Movie

JOIN MovieGenre ON Movie.movieId = MovieGenre.movieId

GROUP BY studioId

HAVING COUNT(DISTINCT genreId) > 2;

SQL.

Create view totalMovies as

SELECT G.genreName, count(\*) AS TotalMvovies

FROM Genre G

left JOIN MovieGenre MG ON G.genreId = MG.genreId

GROUP BY G.genreId ,G.genreName;

**Q5.** *(5 points)* Consider the relation R (A, B, C, D, E), with FDs *F= {AB → C, CD → E, AC → B, DE → A}*. Determine all possible keys for relation R. Prove it.

**Ans: Keys are {ABD}, {CD}, {BDE}.**

**Q6.** *(5 points)* Find a minimal cover of *F = {AB → C, CD → E, A → B, EF → D, BC → EF, A → D}*. Show all steps.

**Step 1:** Decomposition

*{AB → C, CD → E, A → B, EF → D, BC → E, BC → F, A → D}*.

**Step 2:** Remove Redundant F.D

*{AB → C, CD → E, A → B, EF → D, BC → E, BC → F, ~~A → D~~}*.

**Step 3:** Remove Redundant Attribute

*{A~~B~~ → C, CD → E, A → B, EF → D, BC → E, BC → F*}

**Step 4:** Union

*{A → BC, CD → E, EF → D, BC → EF}*

***Minimal Cover***

*{A → BC, CD → E, EF → D, BC → EF}*

**Q7.** *(4 points)* Consider the relation R (A, B, C, D, E), with FDs *F= {AB → C, C → D, DE → A}*. State which of the following decompositions of R relation are lossless decomposition. Prove it.

1. *R1(A, B, E), R2(A, B, C, D)*
2. *R1(A, B, C), R2(C, D, E)*

**Ans:**

**a) Lossless: *R1(A, B, E), R2(A, B, C, D) R1∩R2 → R2-R1 (i.e., AB→CD True)***

**b) Not Lossless: *R1(A, B, C), R2(C, D, E) R1∩R2 → R1-R2 (i.e., C→AB False) and R1∩R2 → R2-R1 (i.e., C→DE False)***

**Q8.** *(6 points)* Consider the relation schema R (A, B, C, D), with FDs *F= {AB → C, CD → A, AD → B}*. Suppose *{AD}* and *{CD}* are the two possible keys of this relation. Identify the best normal form that R satisfies (1NF, 2NF, 3NF, or BCNF). Justify your answer. If R is not in BCNF, decompose it into a set of BCNF relations and show your steps. Indicate which dependencies if any are not preserved by the BCNF decomposition.

**Ans: HNF=3NF; Relation R violate BCNF due to FD1: AB→C,**

**BCNF Schema: R1(A, B, D), R2(A, B, C); FD2: CD*→*A is lost.**

**Q9.** *(15 points)* Draw an ER/EER diagram (using notation discussed in lectures) for the below scenario. Specify all constraints that should hold on to the database and state any assumptions you make.

The dealership sells both new and used cars, and it operates a service facility. A salesperson (SPID, Last Name, First Name) may sell many cars, but each car (CID, Serial No, Make, Model, Color, Year) is sold by only one salesperson. A customer (CusID, Name, Phone No, Address, City) may buy many cars, but each car is bought by only one customer. A salesperson writes a single invoice (IID, Date) for each car he or she sells. A customer gets an invoice for each car he or she buys. A customer may come in just to have his or her car serviced; that is, a customer need not buy a car to be classified as a customer. When a customer takes one or more cars in for repair or service (SID, Name, Hourly Rate), one service ticket (STID, Received Date, Comments) is written for each car. The car dealership maintains a service history for each of the cars serviced. The service records are referenced by the car’s serial number. A car brought in for service can be worked on by many mechanics (MID, Name) and each mechanic may work on many cars. A car that is serviced may or may not need parts (PID, Description, Purchase price, Retail price) adjusting a carburetor or cleaning a fuel injector nozzle does not require providing new parts.

