**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CSE 3330/5330- Database Systems and File Structures**

**Exam #1**

**October 9, 2020**

**Due Date: Monday October 12, 2020 @ 11.59 pm**

|  |  |  |
| --- | --- | --- |
|  | Total points | Earned |
| Multiple choice | 7 |  |
| Short Answers | 20 |  |
| Query Questions | 45 |  |
| Total | 72 |  |

NAME: Suman Thapa Magar

UTA ID: 1001643016

**Multiple Questions:**

1. (1 point) Holds data about one and only one theme in most circumstances, if a table contains more than one theme is needs to be broken up into multiple tables

o table

o Type

o Database

o SQL

**ANS: database**

2. (1 point) Basic building blocks of a relational database

o Entity

o Index Key

o Indexes

o Entity Set

**ANS: entity**

3. (1 point) A key of a relation that consists of two or more columns is a Composite Key

o True

o False

**ANS: True**

4. (1 point) structured query language: data definition and manipulation is SQL

o True

o False

**ANS: True**

5. (3 points) Given a view defined by the following SELECT statement:

SELECT avg(pageCount) as pageAvg, b.pubName

FROM Books b, Publisher p

WHERE b.pubName = p.pubName;

Circle all true statements:

1. This view can be inserted into under the SQL standard
2. This view cannot be inserted into under the SQL standard because it has an aggregate column.
3. This view cannot be updated under the SQL standard because it is over multiple tables.
4. No views can be deleted from under the SQL standard.

**ANS: b and c**

**Short Answer Question (5 points each):**

1. List and define 2 different JOINS used in SQL?

**Ans:** The two different types of JOINS used in SQL are:

1. INNER JOIN:

* It returns records that have matching values in both tables.

1. OUTER JOIN:

* LEFT (OUTER) JOIN: It returns all records from the left table, and the matched records from the right table.
* RIGHT (OUTER) JOIN: It returns all records from the right table, and the matched records from the left table.
* FULL (OUTER) JOIN: It returns all records when there is a match in either left or right table.

1. What are the different types of statements supported by SQL? Define each one.

**ANS:**

The different types of statements supported by SQL are:

1. Data definition Language:

* They are used to define the database structure or table.
* CREATE, ALTER, DROP, TRUNCATE, RENAME

1. Data Manipulation Language:

* They are used for managing data within table object.
* SELECT, INSERT, UPDATE, DELETE,MERGE, LOCK TABLE

1. Data Control Language:

* They are used to give privileges to access limited data
* GRANT, REVOKE, ANALYZE, AUDIT, COMMENT

1. Transaction Control Statement :

* They are used to apply the changes permanently into database
* COMMIT, ROLLBACK, SAVEPOINT, SET TRANSATION

1. Session Control Statement:

* They manage properties dynamically of a user session.
* ALTER SESSION, SET ROLE

1. When and why does the GROUP BY clause get utilized in SQL statement?

**ANS:**

In SQL, the GROUP BY clause is used with the SELECT statement. In the SQL query GROUP BY clause is placed after the WHERE clause and before ORDER BY clause if used any.

GROUP BY clause is used to arrange identical data into groups with the help of some functions.

1. What are aggregate functions in SQL? List 3 aggregate functions.

**ANS:**

Aggregate functions are those functions where the values of multiple rows are grouped together to  form a single value of more significant meaning. The three most common aggregate functions are:

1. COUNT
2. SUM
3. AVG

**Query Questions:**

1. (2 points) Consider the table Driver (licenseNum, firstName, lastName, age), part of a simple driver registration database. Every row of Driver has a unique licenceNum. Write a query in SQL to give the first and last names of all drivers that share a last name with another driver.

**ANS:**

SELECT D1.firstName,D1.lastName

FROM Driver as D1, Driver as D2

WHERE D1.lastName = D2.lastName AND

D1.licenceNum <> D2.LicenceNum

1. Given another relation Voter (voterID, firstName, lastName, district), write a query in SQL to find all (first name, last name) pairs that are associated with a voter in district = ‘42’, and also associated with a driver under the age of 25. Every row of Voter has a unique voterID.
   1. (2 points) Write the query using the INTERSECT operator

**ANS:**

SELECT D.firstName, D.lastName

FROM Driver as D

WHERE D.age<25

INTERSECT

SELECT V.firstName, V.lastName

FROM Voter as V

WHERE V.district = ‘42’;

* 1. (2 points) Write the query WITHOUT using the INTERSECT operator

**ANS:**

SELECT V.firstName, V.lastName

FROM Driver as D, Voter as V

WHERE D.firstName = V.firstName AND D.lastName = V.lastName AND D.age<25 AND V.district=‘42’;

1. (2 points) Given the two relations:

Book(ISBN PK, title, pagecount, pubName FK ref(publisher) )

Publishers(pubName PK, city, state)

Write the SQL creation statement for a virtual view named Bookcities. It should have the title and publishing city of every book.

**ANS**:

CREATE VIEW BookCities

AS SELECT Book.title, Publishers.city

FROM Book, Publishers

WHERE Publishers.pubName = Book.pubName;

1. Use the following UML diagram:



* 1. (1 point) If each band goes on at most 30 tours and can have gone on 0 tours, write in the count for the relation at **(1)**.

**ANS:**

0..30

* 1. (2 points) Write the association class WentOn at **(2)**. When a band goes on tour it gets three attributes a Take, a PlayOrder, and hasMerch.

**ANS:**

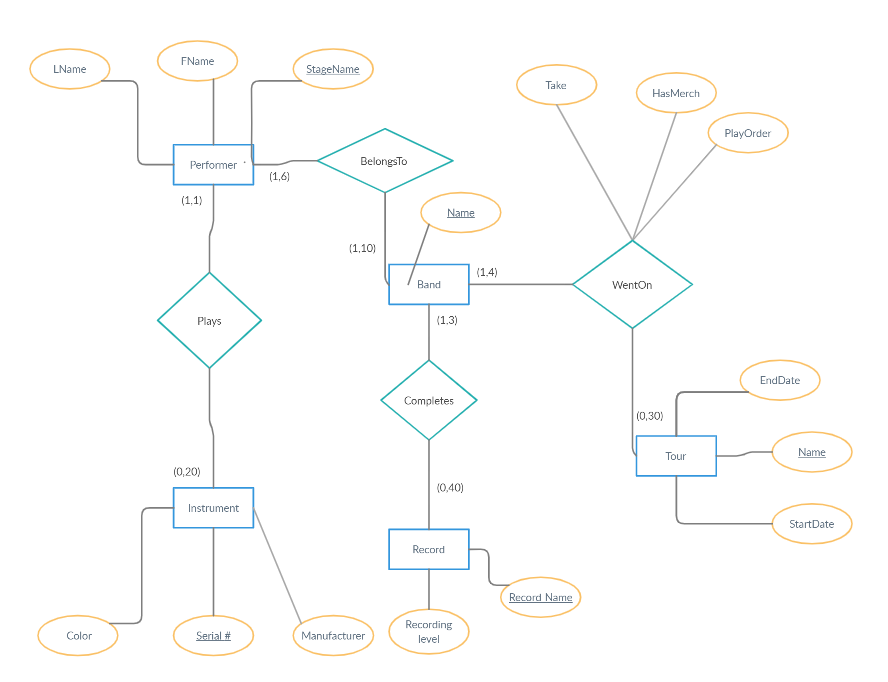
WentOn

Take

PlayOrder

HasMerch

* 1. (5 points) Convert this UML diagram to an ER diagram.



* 1. (5 points) Design the schema for the above UML diagram.

**PERFORMER**

|  |  |  |
| --- | --- | --- |
| FName | LName | StageName |

**BAND**

|  |
| --- |
| Name |

**TOUR**

|  |  |  |
| --- | --- | --- |
| StartDate | EndDate | Name |

**WENTON**

|  |  |  |
| --- | --- | --- |
| Take | PlayOrder | HasMerch |

**RECORD**

|  |  |
| --- | --- |
| RecordName | RecordingLabel |

**INSTRUMENT**

|  |  |  |
| --- | --- | --- |
| Serial# | Manufacturer | Color |

1. For the following questions utilize the following schema and entity set:

Vehicle(VRN, Ma, Mo, Color), Own(VRN, SSN) , Person (SSN, Name, Addr, Phone)



* 1. a. (4 points) Write a query that returns the Name and Phone of all Persons owning VRN=456.

ANS:

SELECT P.Name, P.Phone

FROM Person as P, Own as O

WHERE O.VRN = 456 AND O.SSN = P.SSN;

* 1. b. (4 points) Write a query that returns the Name and Phone of all Persons owning a Suzuki.

**ANS:**

SELECT P.Name, P.Phone

FROM Person as P, Own as O, Vehicle as V

WHERE V.Ma = ‘Suzuki’ AND V.VRN = O.VRN AND

O.SSN = P.SSN;

* 1. c. (4 points) Write a query that returns the Models and Colors of all vehicles owned by someone on Elm.

**ANS:**

SELECT V.Mo, V.Color

FROM Vehicle as V, Own as O, Person as P

WHERE P.Addr = ‘Elm’ AND P.SSN = O.SSN AND O.VRN = V.VRN;

* 1. d. (4 points) Write a query to show the VRN and Mo of each owned Vehicle BY SOMEONE ON 'Birch' and the Name of the Person who owns it.
  2. **ANS:**

SELECT O.VRN, V.MO, P.Name

FROM Vehicle as V, Own as O, Person as P

WHERE P.Addr = ‘Birch’AND P.SSN = O.SSN AND

O.VRN = V.VRN;

* 1. e. (4 points) Write a query that returns pairs of Names for (different) Persons that live at the same Addr.

**ANS:**

SELECT P1.Name, P2.Name

FROM Person as P

WHERE P1.Addr = P2.Addr AND P1.SSN < P2.SSN;

* 1. f. (4 points) Write a query that lists pairs of Persons by SSN and Name that own a Vehicle of the same Ma, Mo, and Color

1. **ANS:**

SELECT P1.SSN P1.Name, P2.SSN, P2.Name

FROM Vehicle as V1, Vehicle as V2, Own as O1,

Own as O2,Person as P1, Person as P2

WHERE P1.SSN < P2.SSN AND

P1.SSN = O1.SSN AND O1.VRN = V1.VRN AND

V1.VRN <> V2.VRN AND

V1.Ma = V2.Ma AND V1.Mo = V2.Mo AND

V1.Color = V2.Color;