**BIS 235 Final Exam**

**Morrese Morrison 10/24/2022**

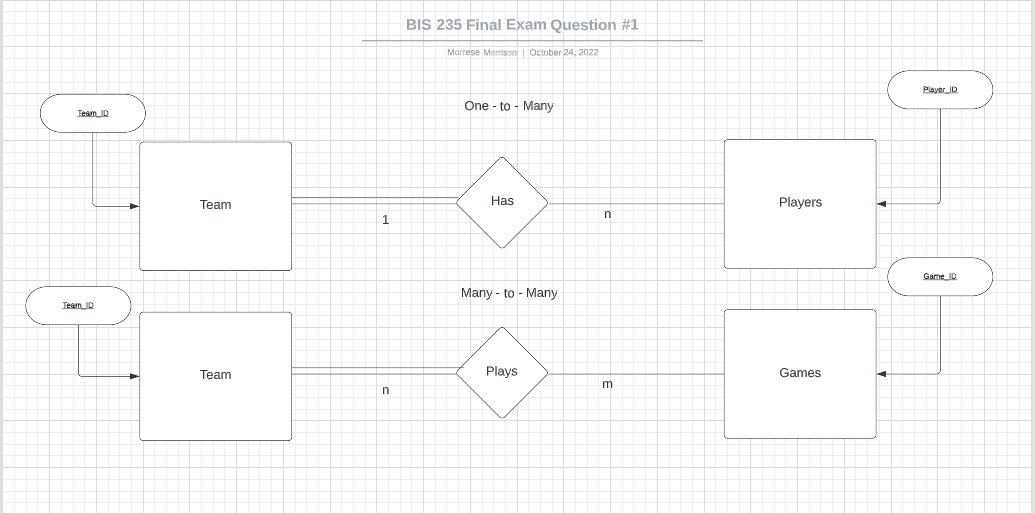
Note: whenever you are asked *why* in the following, be sure for full credit to answer in specific, detailed terms. . Good luck and think hard!

A database is needed to keep track of the teams, games, and players in a sports league. A team has a number of players, not all of whom participate in each game. We need to keep track of the players participating in each game for each team, the positions they played in that game, and the result of each game.

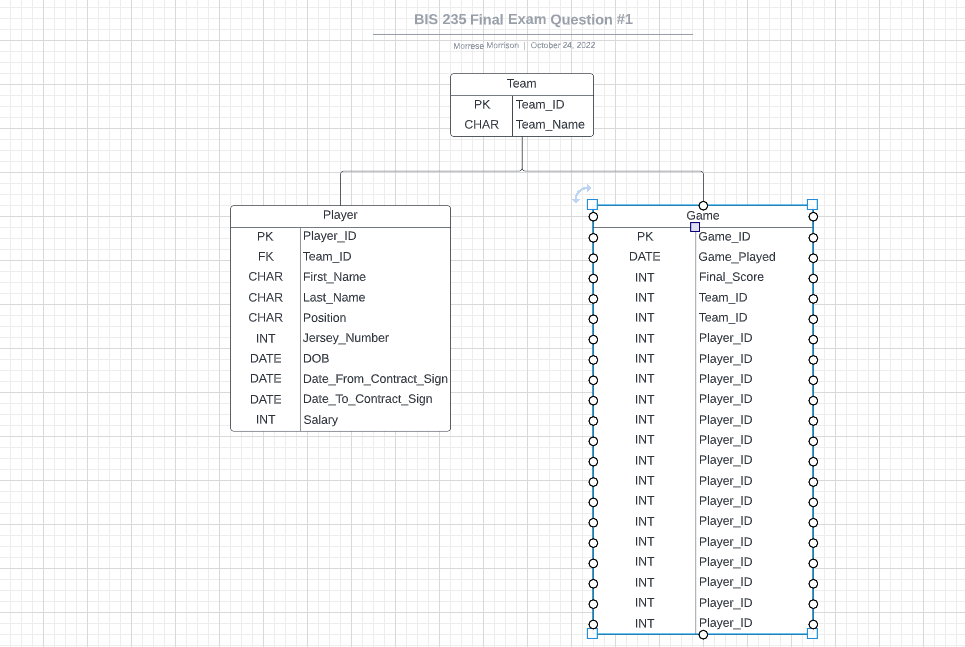
1. **Design the ERD for this database, indicating cardinality and participation and stating any assumptions that you make regarding the problem domain. Indicate all of the attributes for each entity and, as needed, for each relationship. You may do so either directly in your diagram or in a separate table. (20 pts).**

Please see attached photos

1.BIS 235 Final Exam Question #1 Cardinality & Participation



2.BIS 235 Final Exam Question #1 Table Attribute



1. Apply the **Mapped** translation to the following diagram, indicating the resultant relational schema in the space that follows (be sure to indicate all PKs and FKs). (15 pts):



(Note that to streamline your work, all non-key attributes have been eliminated so you only need to deal with primary and foreign keys.)

Mapped Translation

Student(Student\_ID\*)

Score(Score\_ID\*, AssessName)

Absence(Date\*, Student\_ID)

Assessment(AssessName\*, Student\_ID)

1. Apply the **Mapped with Participation Consideration** translation to the following diagram, indicating the resultant relational schema in the space that follows (be sure to indicate all PKs and FKs). (15pts.)



(Once again all non-key attributes have been eliminated. Additionally, entities have been abbreviated with a single letter and relationships with the letters of their participating relationships. Thus, you only need to worry about PKs and FKs here.)

Mapped Translation

A(A#)

B(B#)

C(C#)

D(D#)

Use the following tables for questions 5 through 7:

Student (SID, SName, Phone#, Class#\*)

Class (Class#, Room, Grade, TeacherID\*)

Teacher (TeacherID, TName, Subject)

For questions 5 through 8, write the SQL to answer each question based on the tables indicated above. Make whatever reasonable assumption you would like about data types. Note that to answer the question, you may make use **ONLY** of the info **explicitly** provided in the question itself.

1. Provide a list of the names and phone numbers of all of the students in the 4th grade   
   (i.e. where Grade = 4) (10 pts)

INPUT

**SELECT** SName, Phone#

**WHERE** Grade = 4;

1. What is the name of the teacher that teaches in Room 7A (i.e. where Room = ‘7A’) (5 pts)?  
     
     
   INPUT

**SELECT** TName

**FROM** Class, Teacher

**WHERE** Class.TeacherID = Teacher.TeacherID

**AND** Room = ‘7A’;

1. What are the names of the students who have Jack Black as their teacher (i.e. where in the Teacher table TName = “Jack Black”)? (5pts)

INPUT

**SELECT** SName

**FROM** Class, Teacher, Student

**WHERE** Student.Class# = Class.Class#

**AND** Class.TeacherID = Teacher.TeacherID

**AND** TName = “Jack Black”;

1. How many students are there in the 5th grade (i.e. where Grade = 5) (5 pts)?

INTPUT

**SELECT COUNT (**SName) **AS** ‘Total Students 5th Grade’

**FROM** Student

**WHERE** Grade = 5;

1. Provide the total class size and Class# for each class having more than 25 students. You can identify and refer to the classes by their Class# in so doing. (5 pts)

INPUT

**SELECT COUNT** (**DISTINCT** SID) **AS** ‘Total Class Size’, Class#

**FROM** Student

**WHERE** SID > 25;

Use the following table data to answer questions 9 through 12:

**customers table:**

|  |  |  |
| --- | --- | --- |
| **cust\_id** | **cust\_name** | **cust\_email** |
| 17 | Coyote LLC | ylee@coyote.com |
| 12 | Mouse House | NULL |
| 13 | Rascals | rabbit@wascally.com |
| 15 | Yosemite Place | sam@yosemite.com |

**orders table:**

|  |  |  |
| --- | --- | --- |
| **order\_num** | **order\_date** | **cust\_id \*** |
| 2005 | 2015-09-01 | 17 |
| 2006 | 2015-09-12 | 13 |
| 2007 | 2015-09-30 | 15 |
| 2008 | 2016-10-03 | 17 |
| 2009 | 2015-11-20 | 17 |

**orderitems table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **order\_num\*** | **order\_item** | **prod\_id** | **quantity** | **item\_price** |
| 2005 | 1 | ANV01 | 10 | 5.99 |
| 2005 | 2 | ANV02 | 3 | 9.99 |
| 2005 | 4 | FB | 1 | 10 |
| 2006 | 1 | JP2000 | 1 | 55 |
| 2007 | 1 | TNT2 | 100 | 10 |
| 2008 | 1 | FC | 50 | 2.5 |
| 2009 | 1 | FB | 1 | 10 |
| 2009 | 2 | OL1 | 1 | 8.99 |
| 2009 | 3 | SLING | 1 | 4.49 |

For questions 10 & 11, indicate the actual output that would result from the following query executed against the tables indicated above (in other words, what would the result or “answer” be?):

1. SELECT COUNT(order\_num) FROM orders, customers WHERE cust\_name = "Coyote LLC" AND orders.cust\_id = customers.cust\_id; (5pts)

DESCRIPTION

**Select and count the total amount of orders where the customer name is Coyote LLC.**

OUTPUT

**3**

1. SELECT SUM(quantity\*item\_price) FROM orderitems, orders, customers WHERE orders.order\_num = orderitems.order\_num AND orders.cust\_id = customers.cust\_id AND cust\_name = "Rascals"; (5 pts.)

DESCRIPTION

Sum up the total amount of sales by multiplying the quantity of the item by the item price for any customer that name is Rascals.

OUTPUT

**55**

1. Describe in precise, detailed yet non-technical English what the follow query is asking for: (5pts)  
   SELECT cust\_name from customers, orders WHERE customers.cust\_id = orders.cust\_id AND DATEDIFF(NOW(), order\_date) < 365;

DESCRIPTION

**Find customer names who placed an order in less than a year.**

OUTPUT

**Empty Set – all order dates are greater than the last year due to orders being in 2015, 2016**

1. Describe in precise, detailed yet non-technical English what the following query is asking for: (5 pts.)

SELECT order\_num,

count(order\_item)

FROM orderitems

GROUP BY order\_num

HAVING(count(order\_num)>2);

DESCRIPTION

**Find order numbers that have more than 2 items.**

OUTPUT

2