**Metropolitan State University**

**ICS 311-02**

**Final Exam**

**Total Points: 100**

**Name: Nalongsone Danddank Date: 12/09/2020**

# Question 1 (Total 10 Points): Multiple Choice Questions:

**Circle ONLY ONE correct answer. (If more than one answer is circled then your answer will not be considered even if it includes the correct answer).**

1. An ER Diagram consists of all the following, except: Answer: C
   1. Entity b. Entity Sets
2. Index d. Relation
3. In case of transaction failure due to illegal operation by the transaction (e.g., division by zero), *to set data item to old value*, which of the following will be needed? Answer: a
   1. Undo operations b. Redo operations

c. Undo and Redo operations d. None of the above.

1. In order to recover from a crash, the recovery manager recovers committed transactions that committed after the last checkpoint by: Answer: b

a.Undoing operations b. Redoing operations

c. Doing nothing

1. A data dictionary identifies all of the following, except: Answer: c
   1. Attribute names b. Attribute data types

c. Table statistics d. Attribute values

1. The following rule must be maintained by the DBMS to ensure data consistency: Answer: d
   1. Entity integrity constraints c. Referential integrity constraints
   2. Attribute domain constraints d. All of the above
2. To judge on the correctness of a given schedule for the execution of concurrent transactions we use: Answer: c
   1. Wait-for graph b. Referential integrity constraint

c. Serializability/precedence graph d. Checkpoint

1. All of the following are deadlock prevention protocols, except: Answer: c
   1. Wait/die Protocol b. Wound/wait protocol
2. Concurrency control
3. All of the following use index search keys, except: Answer: c
   1. Dense Index b. Clustered index

c. Hash based index d. B+-tree index

1. All the following are valid cardinalities in an ERD, except: Answer: c
   1. One-to-one b. Many-to-one

c. Single-to-Multiple d. One-to-Many

1. All the following are valid forms of normalization, except: Answer: c
   1. BCNF b. Third form

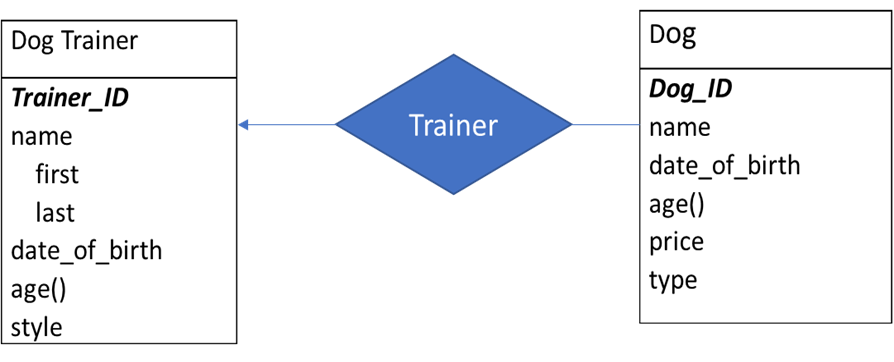
c. Schedules d. First form

# Question 2 (Total 10 Points): True/False Questions:

**For each of the following statements, circle T (True) or F (False)**

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | The combination of primary keys of the participating entity sets  forms a super key of a relationship set Answer: F | **T** | **F** |
| 2 | When using the two-phase locking protocol, two transactions can hold exclusive locks on the same item at the same time. Answer: F | **T** | **F** |
| 3 | The recovery manager ensures the Atomicity property of a transaction by undoing all operations of aborted transactions. Answer: T | **T** | **F** |
| 4 | A hash index is more efficient than a B+ tree index in evaluating a query with an equality condition. Answer: T | **T** | **F** |
| 5 | If a salesperson attempts to enter information in INVOICE table for a customer not in the CUSTOMER table, the database will typically generate an error message. This message indicates that a referential integrity constraint has been violated. Answer: T | **T** | **F** |
| 6 | Triggers can only be activated *after* an update/insert/delete into a relation, but not *before*. Answer: F | **T** | **F** |
| 7 | Exclusive mode lock allows another transaction to make an update to the same data item Answer: F | **T** | **F** |
| 8 | The crash recovery algorithm consists of two phases, the first is the Redo phase and the second is the Undo phase. Answer: T | **T** | **F** |
| 9 | A foreign key is an attribute that uniquely identifies each row in the table where it is defined Answer: F | **T** | **F** |
| 10 | Assuming that each transaction preserves database consistency when runs in isolation, then the serial execution of a set of transactions is guaranteed to preserve database consistency. Answer: T | **T** | **F** |

# Question 3 (Total 15 Points): Database Design:

****

## 3.1 (6 Points) Given the ERD above, write down DDL(Create SQL) needed to convert this ERD into a relational schema. Make sure to show all primary and foreign keys.

Create database dog\_trainer\_db;

Use dog\_trainer\_db;

Create table Dog\_Trainer(

Trainer\_ID integer,

fname varchar(20) not null,

lname varchar(20) not null,

date\_of\_birth Date,

age integer,

style varchar(20),

primary key(Trainer\_ID));

Create table Dog(

Dog\_ID integer,

name varchar(20),

date\_of\_birth Date,

age integer,

price float,

type varchar(20),

Trainer\_Id integer,

primary key(Dog\_ID),

foreign key(Trainer\_Id) references Dog\_Trainer(Trainer\_ID)

on delete set null);

delimiter $$

create trigger trg\_Dog\_Trainer

after insert on Dog\_Trainer

for each row

begin

update Dog\_Trainer set age = TIMESTAMPDIFF(YEAR, date\_of\_birth, CURDATE());

end $$

delimiter ;

delimiter $$

create trigger trg\_Dog

after insert on Dog

for each row

begin

update Dog set age = TIMESTAMPDIFF(YEAR, date\_of\_birth, CURDATE());

end $$

delimiter ;

## 3.2 (4 Points) Write Insert statements to fill at least 2 records each into all the tables.

Insert into Dog\_Trainer(Trainer\_ID, fname, lname, date\_of\_birth, style) values(1, “Robert”, “Smith”, “12-07-1989”,”Jumbing”),(2, “Jimmy”, “Hou”, “04-05-1990”, “Hubby”);

Insert into Dog(Dog\_ID, name, date\_of\_birth, price, type, Trainer\_Id) values(1, “Keytie”, “10-12-2016”, 450.95, “short dog”, 1),(2, “Kumbee”, “10-12-2019”, 199.50, “Fat dog”, 2);

## 3.3 (5 Points) Write down the SQL query needed to find all the dogs (name) and Trainer Style, that are trained by trainers named “Robert Smith”.

Select d.name, t.style

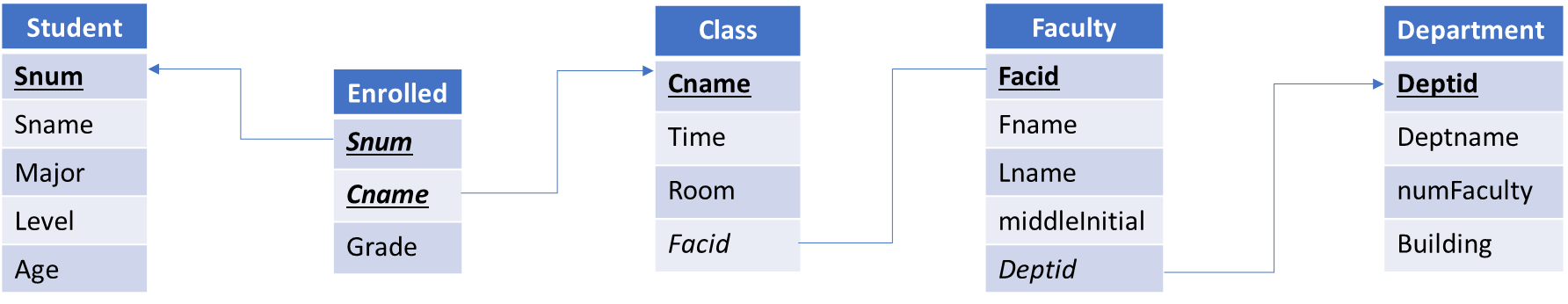
from Dog d join Dog\_Trainer t

on (d.Trainer\_Id = t.Trainer\_ID)

where t.fname = “Robert” and t.lnaem = “Smith”;

# Question 4 (Total 40 Points): SQL Queries:

Given the following schema. Primary keys are **bold/underlined** and foreign keys are in *italics*. The meaning of the relations is straightforward.



Write SQL queries to answer the following questions:

**4.1) (5 Points) Find the name and age of the youngest student.**

## Select Sname, Age from Student order by Age limit 0,1;

**4.2) (5 Points) Find name, level and age of students who are not enrolled in any class. (Hint: use NOT IN)**

## Select s.Sname, s.Level, s.Age from Student s where s.Snum NOT IN (select e.Snum from Enrolled e);

## 4.3) (5 Points) Find the Student Name, Age, Faculty Name, Department Name of all Postgrad students (level = PG) who are enrolled in a class taught by faculty ‘Charles A Gibbs’

*Note: Pay attention to middleInitial too*

## Select s.Sname, s.Age, concat(f.Fname, “ ”, f.middleInitial, “ ”, f.Lname) name, d.Deptname

From Student s, Faculty f, Department d, Enrolled e, Class c

where s.Snum = e.Snum and s.Level = “PG” and e.Cname = c.Cname and c.Facid = f.Facid and f.Deptid = d.Deptid and f.Fname = “Charles” and f.Lname = “Gibbs” and f.middleInitial = “A”;

## 4.4) (5 Points) Create a view called ‘Faculty\_Teachers’. This view includes a list of all class names along with First and Last Name of the faculty who teach this course. (i.e. view output includes cname, fname and lname), only where *faculty teach in building ‘London’*.

Create view Faculty\_Teachers as

Select c.Cname, f.Fname, f.Lname

from Faculty f, Class c, Department d

Where f.Facid = c.Facid and c.Deptid = d.Deptid and d.Building = “London”;

## 4.5) (10 Points) Create a stored procedure that takes two inputs: a faculty identifier (facident) and number of bonus points (bonuspt). The procedure then updates the Enrolled table by adding the bonus points to the grades of all courses that are taught by faculty with the input facident.

## *Note: Do not forget to use delimiter.*

## *Hint: Consider using subquery for WHERE clause in UPDATE statement*

delimiter $$

create procedure prc\_update\_faculty\_bonus(in facident integer, in bonuspt float)

begin

update Enrolled e set e.Grade = e.Grad + bonuspt where e.Cname = (select c.Cname from Class c where c.Facid = facident);

end $$

delimiter ;

## 4.6) (10 Points) Create a trigger to increment the numFaculty attribute of the Department table after a new row is inserted in the Faculty table.

## *Note: Do not forget to use delimiter.*

delimiter $$

create trigger trg\_incr\_numFac\_dept

after insert on Faculty

for each row

begin

update Department set numFaculty = numFaculty + 1;

end $$

delimiter ;

# Question 5 (Total 15 Points): Transactions and Crash Recovery

**5.1) (5 Points) Consider following Serial Schedule. Assume initial values as:**

**A=$2000 B=$3000**

**Consistency property of transaction states that the Sum of A + B should remain 5000 for this schedule. Write down results of each step in the ‘Results’ column. Based on the result, determine if the sum of A and B stays $5000.**

|  |  |  |
| --- | --- | --- |
| T1 | T2 | **Results** |
| read(A) |  | **2000** |
| A :- A - 100 |  | **1900** |
| write (A) |  | **1900(A)** |
|  | read(A) | **1900** |
|  | temp:= A \* 0.1 | **190** |
|  | A := A - temp | **1710** |
|  | write(A) | **1710(A)** |
| read(B) |  | **3000** |
| B := B + 100 |  | **3100** |
| write B) |  | **3100(B)** |
| commit |  | **Total A+B = 5000** |
|  | read(B) | **3100** |
|  | B := B + temp | **3290** |
|  | write(B) | **3290(B)** |
|  | commit | **Total A+B = 5000** |

## 5.2) (5 Points) Following is an example of banking transaction. In each case of (a), (b), (c) transaction T0 hits a failure. Based on the step in each scenario, an Undo or Redo, or a combination of recovery steps are performed.



What will be the value of (a), (b), (c)?

List what recovery actions were performed for each step (Undo/Redo).

1. A, B and C

A = 450

B = 980

C = 1600

1. A, B and C

A = 450

B = 980

C = 1700

1. A and B

A = 800

B = 765

## 5.3) (5 Points) Match the following Index names with their definitions:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Index Definition** |  |  | **Index Name** |
| A | This type of index is a rooted tree where all paths from root to leaf are of the same length |  | 1 | Dense Index |
| B | In this type of index, index record appears for every search-key value in the file |  | 2 | Sparse Index |
| C | In this type of index, a bucket is a unit of storage . Records with different search-key values may be mapped to the same bucket |  | 3 | B+ Tree Index |
| D | This type of index is created If order of data records is the same as, oe close to, order of data entries |  | 4 | Hash Index |
| E | This type of index uses less space and less maintenance overhead for insertions and deletions. It contains index records for only some search-key values. |  | 5 | Clustered Index |

A = 3

B = 1

C = 4

D = 5

E = 2

# Question 6 (Total 10 points): Concurrency Control

## 6.1) (5 points) For following schedule, draw the serlializability graph (also known as precedence graph or conflict graph) and determine:

## *a) Is the schedule conflict serializable or not*

## *b) Explain why or why not*

|  |  |  |
| --- | --- | --- |
| T1 | T3 | T2 |
| R(x) |  |  |
|  |  | R(z) |
| R(z) |  |  |
|  | R(x) |  |
|  | R(y) |  |
|  | W(x) |  |
|  |  | R(y) |
|  |  | W(z) |
|  |  | W(y) |

1. The schedule is  *conflict serializable*
2. *Because there have no circle in the graph*

## 6.2) (5 points) Draw the Wait-for-Graph and determine:

## *a) Is there a deadlock or not*

## *b) Explain why or why not*

1. There is a deadlock
2. Because there have a circle in the Wait-for-Graph (T19 -> T18 -> T20 -> T19)