CST363 Spring 2018a Exam 1

Name _____Christopher Holmes_____

Directions:

Edit this file to include your answers to questions and then upload and submit the completed file in DOCX or PDF format to iLearn into the Exam1 assignment.

There is no time limit for this exam but you must submit by 11:50 pm on Tuesday Jan 23.

You may use your textbook and other class resources, but you must not get or give help from others.

Your answers should be well written sentences (correct spelling, grammar, logical reasoned) when appropriate.

Sign the honor pledge.

I certify that I have neither given nor received unauthorized help in taking this exam. This exam is my own work.

_____< sign your name here>_____

Question	Points	Your Score
1	5	
2	5	
3	5	
4	5	
5	5	
6	5	
7	5	
8	5	
9	5	
10	6	
11	5	
12	5	
13	6	
14	6	
15	5	
16	6	
17	4	
18	4	
19	4	
20	4	
Total	100	

1. Explain briefly some of the update problems when data for multiple entities is stored in a single table – much like a single spreadsheet. Give an example to illustrate the problems.
    a. You can run into issues when the data is stored in a single table on multiple accounts. You will run into issues when trying to insert data when you only want to insert for one entity, but there are the other entities in the table that shouldn't be there. Modifying can cause issues when there is duplicate data. You could possibly update only some of the data, and then that would cause inconsistencies in your data where some is updated and some is not. Deletion will also cause problems. Because you have multiple entities in a single table, when you only want to delete one entity, the other ends up getting deleted too because it's on the same table.

2. Explain the difference between a client/server with a database server versus a web-based system that uses a database server. [go back and reread chapter 1 if necessary]
    a. In a client server model, for the user to be able to access data, they require a piece of software that is able to talk to the database server. This piece of software requires the user to be able to know SQL so they can write the queries to get the data they need. There is also the ability to use a data API for the database, where you would then have to use the programming language that works with the API.
    b. In a web-based system, the user does not need any special software. The user is interacting with the database through their web browswer that is running a web application. This web application is going to be designed specially to allow the end user to do what they need, and it requires very little knowledge of the setup of the database or SQL, for the user to be able to do what they need to do.

3. Schema information (names of database schemas, table names, column names and data types, as well as status information about the DBMS server) are stored in tables. This information can be accessed using the MySQL SHOW command, a SELECT statement, and the information can be displayed in the workbench. This information is often called "metadata" (data about data) and is also referred to as the "system catalog".

    This link https://dev.mysql.com/doc/refman/5.7/en/information-schema.html refers to a page in the MySQL reference that lists all the catalog tables (there are quite a few). Catalog tables are in a special database with a schema name of "information_schema".

    What catalog table and what column in the table in the "information_schema" would you look at to see if a column allows null values?

    -To see if a column allows for null values, you would look in the INFORMATION_SCHEMA.COLUMNS table, and you would look at the column name 'IS_NULLABLE'. If this returns YES, then you can have a null value, if it returns no, then you cannot.

For questions 4-13 use the following table schema. An sql script is available that will create the tables and load with sample data to assist you in answering questions 4-13.

Table: Student

StudentID	LastName	FirstName	GPA	Major	TotalCredits

Table: Enrollment

StudentID	**CourseId**	Semester	Grade	Units

Table: Course

CourseId	Name	Section	Department	Instructor

Write an SQL select that can answer the question.

4. Report on all the students (LastName, FirstName) who are majoring in Biology and have a GPA between 3.0 and 3.5 The result should be sorted by LastName and FirstName.

```
select lastname, firstname
from STUDENT
where major = 'Biology' and (gpa between 3.0 and 3.5)
order by lastname, firstname
```

5. Summarize the database by major. For each major show the number of students with that Major and the average GPA of students in that Major.

```
select major, count(studentid) as num_students, AVG(gpa)
from student
group by major
```

6. List all the students (LastName, FirstName) who took the course with name “Discrete Math”.

```
select distinct lastname, firstname
from STUDENT
```

```
join ENROLLMENT on STUDENT.studentid = ENROLLMENT.studentid
where courseid = (select courseid from COURSE where COURSE.name = 'Discrete
Math')
order by lastname, firstname
```

7. List all students (LastName, FirstName) took “Discrete Math” and got a grade of “B” or “A”.

```
select distinct lastname, firstname
from STUDENT
join ENROLLMENT on STUDENT.studentid = ENROLLMENT.studentid
where courseid = (select courseid from COURSE where COURSE.name = 'Discrete
Math')
and grade = 'A' or 'B'
order by lastname, firstname
```

8. List the number of students enrolled in each course. If a course has no students enrolled the course with a count of 0 should appear in the result. The result should have CourseID, Name and count in the result set and be in CourseId sequence.

```
select COURSE.courseid, name, count(studentid) as count_students
from COURSE
join ENROLLMENT on ENROLLMENT.courseid = COURSE.courseid
group by COURSE.courseid
```

9. Find students enrolled in course name=’Calculus 2’ Semester=’2018Spring’ who have taken and passed with a grade of C or higher the required prerequisite of course ’Calculus 1’.

```
select STUDENT.studentid
from STUDENT
join ENROLLMENT on STUDENT.studentid = ENROLLMENT.studentid
join COURSE on ENROLLMENT.courseid = COURSE.courseid
where COURSE.name = 'Calculus 2' and ENROLLMENT.semester = '2018Spring' and
STUDENT.studentid in(
select ENROLLMENT.studentid
from ENROLLMENT
join COURSE on ENROLLMENT.courseid = COURSE.courseid
where COURSE.name = 'Calculus 1' and ENROLLMENT.grade in ('C', 'B', 'A'))
```

10. There are 3 professors who teach Calculus 1: Smith, Lupin and Fenwick. We want to compare the average grade of students in Calculus 2 based on the teacher in Calculus 1. If a student took Calculus 1 but did not take Calculus 2 don't count them. The result should have

InstructorCaculus1	NumberTakingCalculus2	AverageGradeCalculus2
Smith		
Lupin		
Fenwick		

```
select instructor as 'InstructorCaculus1', count(grade) as 'NumberTakingCalculus2', AVG(
case grade
        when 'A' then 4
   when 'B' then 3
   when 'C' then 2
   when 'D' then 1
   when 'F' then 0
end) as 'AverageGradeCalculus2'
from course
join enrollment on course.courseid = enrollment.courseid
where course.name = 'Calculus 1' and studentid in (
select studentid
from enrollment
join course on enrollment.courseid = course.courseid
where name = 'Calculus 2')
group by instructor
```

11. What SQL select would you use if you need to print a transcript for a student given a student id? The transcript should be in order by year and semester and should include course name, grade, units.

```
select name, grade, units, semester
from enrollment
join course on enrollment.courseid = course.courseid
where studentid = 5430
order by semester
```

12. A student wants to enroll in Math 170 Discrete Mathematics. If the student has already taken the course and passed, or taken the course 2 times and failed, they cannot enroll again. Supposed the student id is 123456, what select can you use to determine if the user can enroll

in this course? The select statement should return the number of times student 123456 has taken the course and the highest grade received in the course.

```
select count(courseid) as 'NumTries', max(grade)
from enrollment
where studentid = 123456 and courseid = 40132
group by courseid
```

13. Write an sql update statement that changes the grade for student with last name Last14 who has taken course Discrete Math. change the grade from B to A. Do this with a single update statement using the value "Last14" for lastname, course name "Discrete Math". Verify that the current grade is a B using a WHERE predicate.

```
update enrollment
join student on enrollment.studentid = student.studentid
set grade = 'A'
where lastname = 'Last14' and grade = 'B';
```

14. Read about slow indexes part 1 at "Use the Index Luke" website http://use-the-index-luke.com/sql/anatomy/slow-indexes Indexes will usually improve performance but they will not necessarily make processing all SELECT statements quick. Even with indices there is work to be done.
    a. Describe in your own words, the steps that the DBMS has to do to retrieve data records in a table using an index.

    The three steps that a DBMS has to do to retrieve a record are, traverse the entire tree, follow the chain, and returning the data. It has to traverse the tree to get the indexes, follow the chain to make sure there all entries are touched, and then finally to return the data because that is what you were trying to do in the first place.

    b. Assume that the index is not unique (multiple rows have the same column value such as a foreign key column). Assume that all levels of the index except the sequence set (also called leaf nodes) are kept in memory of the DBMS server. Assume that each random I/O takes 0.001 seconds (1 millisecond). Explain what the author means by "slow indexes".

    Because there is the possibility that the entries that have the same index could possibly have more entries below that are related to the index, so to do a full lookup, the DBMS must follow the entire chain to verify that there aren't anymore entries below that could be needed.

c. If an index is unique (like a primary key index), how does that change your answer to part b.

> If an index is unique, that means there will be only one entry with that index. Because there is only one entry, it does not have to follow the entire chain to verify there aren't any entries below it that needs.

15. Read about how indices are used when the SELECT has a LIKE predicate. http://use-the-index-luke.com/sql/where-clause/searching-for-ranges/like-performance-tuning Assuming there is an index on the Name column, which of the following predicates will use the index. Explain your reasoning.
    a. where name like 'United %'
    b. where name like '% States'

> A will use the index because it has an access predicate. B only has a filter predicate, so the entire table will still be scanned before anything filtering is done.

16. The table world. country has an index on Code column because Code is the primary key. Create an index on the Name column using the sql statement

```
create index name_index on world.country(name);
```

Use MySQL Workbench to show the query plan for the following queries.

```
select * from world.country where code='USA' and name='United States';
select * from world.country where code='USA' or name='United States';
```

a. Describe how the DBMS will process the first SELECT statement that uses "and". Which index is used and why do you think this index is used rather than the index on "name"?

> The way the DBMS found the result in this one was it selected a single row from the country table that used the code index. The index that is being used in this select statement is the code index. It is being used because the cost to find the result is faster using the code index rather than the name.

b. Described how the DBMS will process the second SELECT using "or". What does "using union" mean in the explain output?

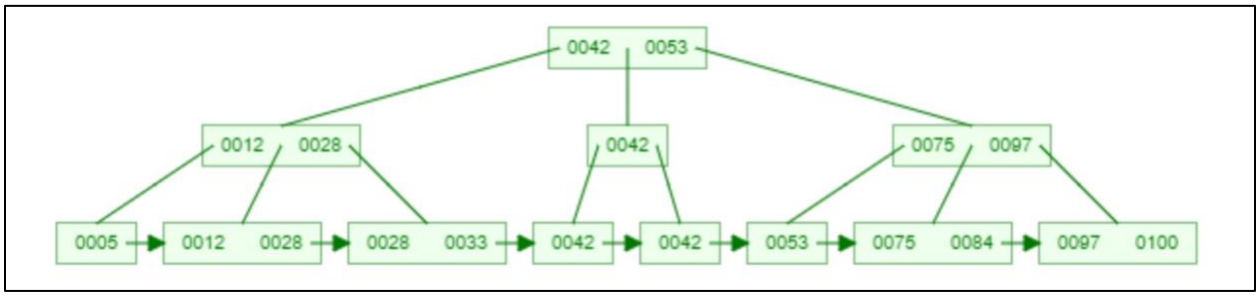
> This result is returned by doing a union of the two indexes, the code index and the name_index that we created. Because we are using an or, the cost to return this result is much higher than the and, and that is why it requires the two indexs.

Use the B-tree simulator at http://www.cs.usfca.edu/~galles/visualization/BPlusTree.html

Set Max Degree to 3. Insert the following values in the order:

12 100 97 5 42 42 53 84 75 33 28 28

The B-Tree should look like



Each rectangle represents a node in the tree. The top node is the "root" and the bottom nodes are the sequence set.

17. Doing the select statement with a predicate column***=33***, what nodes must be read to find the record with value 33?
    a. It reads the root node to see that 33 is less than 42, so it goes to the left to access the node that has 12 – 28. It sees that 33 is greater than 28, so it will go down the right branch to reach the node that contains 28-33, finally finding 33

18. Which nodes must be updated or new nodes created when doing an insert with value 78?
    a. To insert the value 78, we must update the node that contains currently 84 – 97. This node will contain the values 78 – 97, and a new node will be created below that containing 78.

19. With a max degree of 3 (each node has 3 child nodes below it), approximately how many levels would be needed to index a file with 1,000,000 records?
    a. You would need approximately 200000 levels to index with max degree of 3
    2T-1 where T is degree 1000000/(2T-1)

20. Suppose the max degree was 100 instead of 3. Now many levels are required?
    a. You would need approximately 50225 levels to index with max degree of 100
    2T-1 T is degree 1000000/(2T-1)

Course feedback. I would very much appreciate your feedback on the course so far. I use this information to constantly improve the class. Your answers here do not affect your score.

1. Are you satisfied with the help available in the course and the feedback that you receive on assignments and quizzes from instructor, TA and fellow students? What changes (if any) would you like to see?
    a. The feedback that has been provided in this course has been great. Assignments are graded in a very timely manner, and when points are deducted, there is a very clear explanation of why points were deducted. Very helpful in understanding how to improve and do it right next time.

2. What is your opinion of the Murach MySQL textbook. Do you think that the book is well written, easy to understand, relevant to your career goal? Did you find anything missing that you wished had been included in the book?

    Rating: 1 (lowest) – 5 (highest)

    Why? I think that the book is fairly good at what it is trying to get conveyed. I had taken a previous database class at my community college that used a different book, cannot remember the name of it unfortunately, that I thought was slightly better. The examples seemed to be more clear in that book and the concepts seemed to be easier to understand with the examples and text.

3. Rate the following topics and resources as helpful or not in your learning.

	Helpful	Not Helpful
Videos on SQL	x	
Assignments 2 and 3 on SQL	x	
Assignment 4 on index structures	x	
Assignment 6 on query plans	x	

4. Other comments positive or negative you have about the course so far?

    I am really enjoying this class. This has been the first class that has really provided a programming challenge for me so far and I am really enjoying it. I really appreciate the detailed feedback that is provided on the assignments and quizzes when points are deducted. I also really appreciate the promptness of grading because in previous classes it could be weeks before

we had a grade on an assignment, and with as fast as these classes move, we're way past that weeks assignment by the time we were getting a grade.