

Points missed: \_\_\_\_\_

Student's Name: \_\_\_\_\_

Total score: \_\_\_\_\_ /50 points

East Tennessee State University – Department of Computer and Information Sciences  
CSCI 2910 (Tarnoff) – Server/Client Side Programming  
TEST 2 for Spring Semester, 2007

**Read this before starting!**

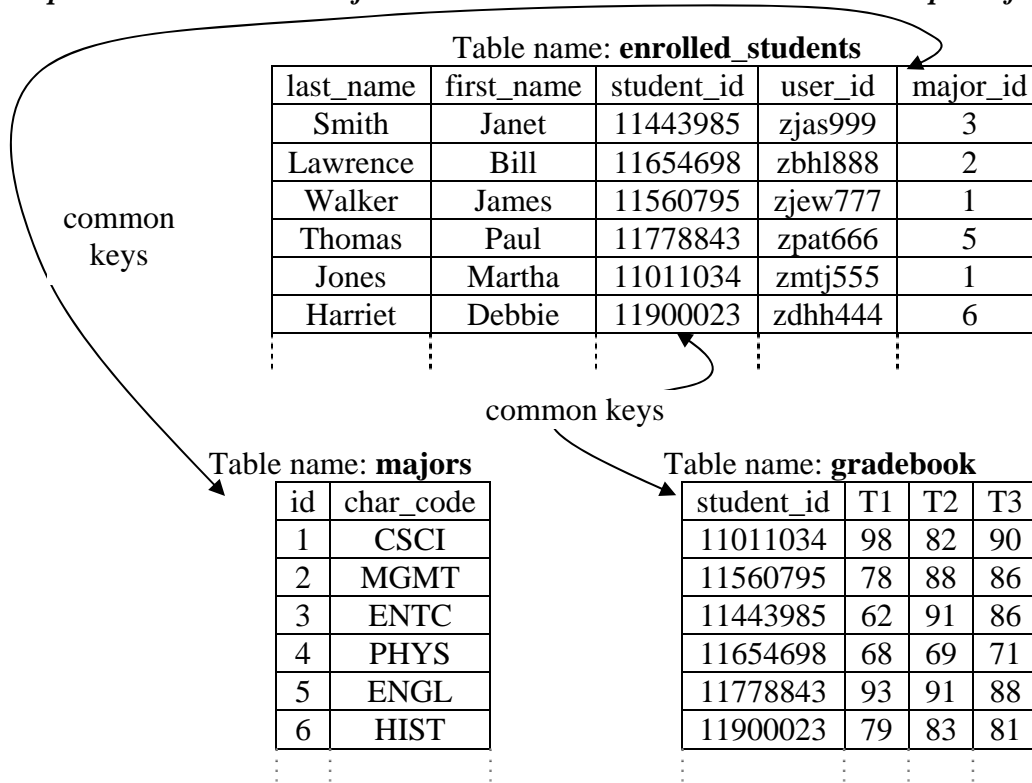
- The total possible score for this test is 50 points.
- This test is ***closed book and closed notes***.
- A summary of MySQL commands can be found at the bottom of this page.
- ***Please turn off all cell phones & pagers during the test.***
- You may ***NOT*** use a calculator. Complex numeric calculations may be left in the form of an expression.
- All answers must be placed in space provided. Failure to do so will result in no credit for answer.
- If you perform written work on the back of a page in this test, indicate that you have done so in case the need arises for partial credit to be determined.
- Statement regarding academic misconduct from Section 5.7 of the East Tennessee State University Faculty Handbook, June 1, 2001:

"Academic misconduct will be subject to disciplinary action. Any act of dishonesty in academic work constitutes academic misconduct. This includes plagiarism, the changing of falsifying of any academic documents or materials, cheating, and the giving or receiving of unauthorized aid in tests, examinations, or other assigned school work. Penalties for academic misconduct will vary with the seriousness of the offense and may include, but are not limited to: a grade of 'F' on the work in question, a grade of 'F' of the course, reprimand, probation, suspension, and expulsion. For a second academic offense the penalty is permanent expulsion."

SQL/MySQL Syntax:

- INSERT INTO tablename (fieldname [, fieldnames]) VALUES (value [, values])
- DELETE FROM tablename WHERE fieldname=value
- UPDATE tablename SET fieldname=value WHERE fieldname=value
- USE database
- SHOW TABLES
- CREATE TABLE tablename (field1 data\_type [NULL/NOT NULL], field2 data\_type [NULL/NOT NULL], ...)
- DROP TABLE tablename
- SELECT [ ALL | DISTINCT ] \*| COLUMN1[, COLUMN2 ] FROM TABLE1 [ , TABLE2 ] WHERE [CONDITION1 | EXPRESSION1][ AND|OR CONDITION2 | EXPRESSION2 ] ORDER BY fieldname [, fieldnames] [ASC|DESC] LIMIT [start,] size

All problems on this test refer to the three tables shown below that are part of a relational database.



Here are some additional comments about the above table. First, the tables are only partial tables. Assume there is more data. Second, enrolled\_students and gradebook could probably have been the same table, but they were kept separate for reasons of security, i.e., not wanting people who had access to the roll also have access to the gradebook. Third, the table majors could be made unnecessary by simply using the char\_code directly for the major\_id in enrolled\_students. The problem with that is that if a char\_code gets changed, you would have to go through all of the enrolled students and manually make the change instead of making the change in one place, i.e., the majors table.

1. What score did Martha Jones get on Test 2 (T2)? (2 points)

From the enrolled\_students table, we see that Martha Jones' student\_id is, 11011034. Using this student\_id, we see that her grades in the gradebook table are in the first row. For T2, she received an **82**.

2. Which field would serve best as the primary key for the **enrolled\_students** table? (Note that there are actually 2 answers, but one is better than the other.) (2 points)

Both student\_id and user\_id are unique fields for all users. Since the majority of references to this table and others in the database will be through the *student\_id*, it is probably a better choice for the primary key.

3. Which field(s) would serve best as the primary key for the **gradebook** table? (2 points)

The only unique field in the gradebook table is *student\_id*.

4. If we wanted to add a field to identify the student's level, i.e., freshman, sophomore, junior, or senior, which table should we use? (2 points)

The only table that identifies students information is the enrolled\_students table. There isn't any student information in majors, and it is highly unlikely that all students would be listed in gradebook.

- a.) enrolled\_students      b.) majors      c.) gradebook      d.) need to add a fourth table

5. If we wanted to add a field to identify properties of the class such as instructor name, meeting time, or assigned room, which table should we use? (2 points)

There is no table representing course information. The closest is gradebook, but that table should only contain a list of the students associated with that class and their corresponding grades. A new table should be added to hold course information.

a.) enrolled\_students      b.) majors      c.) gradebook      **d.) need to add a fourth table**

6. What SQL data type would you assign to the field **id** in the table **majors**? Be as specific as you can. There will never be more than 256 majors. (2 points)

The clue was that there will never be more than 256 majors. Therefore, an integer from 0 to 255 would be sufficient to uniquely identify different majors. This means a TINYINT will be the best type.

a.) BIT      **b.) TINYINT**      c.) INTEGER      c.) BIGINT      d.) FLOAT      e.) DATE  
f.) DATETIME      g.) YEAR      h.) CHAR      i.) VARCHAR      j.) TEXT      k.) ENUM

7. What SQL data type would you assign to the field **char\_code** in the table **majors**? Be as specific as you can. This field will always have exactly 4 characters. (2 points)

The clue was that there will always be exactly 4 characters. This means a CHAR(4) will be the best type.

a.) BIT      b.) TINYINT      c.) INTEGER      c.) BIGINT      d.) FLOAT      e.) DATE  
f.) DATETIME      g.) YEAR      **h.) CHAR**      i.) VARCHAR      j.) TEXT      k.) ENUM

8. Write the SQL statement to create the table **majors**. (3 points)

**CREATE TABLE majors (id TINYINT, char\_code CHAR(4));**

Note that even if you got the wrong answers in 6 and 7, the data types you used to answer this question must match what you answered for problems 6 and 7.

9. Complete the partial MySQL select statement below in order to display all fields from all records contained in the table enrolled\_students *sorted by last\_name then by first\_name*. (3 points)

***select \* from enrolled\_students order by last\_name, first\_name;***

10. Complete the partial MySQL select statement below in order to display all fields from *records 2, 3, and 4* of the table enrolled\_students. (Note that the record numbers identifies the position of the record within the table, the first record being position 0.) (3 points)

***select \* from enrolled\_students limit 2, 3;***

11. Complete the partial MySQL select statement below in order to display all fields of the single record contained in enrolled\_students with a *user\_id equal to "zmtj555"*. (3 points)

***select \* from enrolled\_students where user\_id="zmtj555";***

12. Complete the partial MySQL select statement below in order to display all fields of records contained in enrolled\_students with a *student\_id* containing '10' as the second and third digit. (3 points)

*select \* from enrolled\_students where student\_id like '\_10%';*

13. Complete the partial MySQL select statement below in order to join the tables enrolled\_students and gradebook to display the students' names and their corresponding test grades. (4 points)

*select last\_name, first\_name, T1, T2, T3 from enrolled\_students, gradebook where  
enrolled\_students.student\_id = gradebook.student\_id;*

14. Write the SQL statement to output the student\_id and the result of the final grade calculation  $(T1*0.4)+(T2*0.2)+(T3*0.4)$  from the table gradebook. (4 points)

*select student\_id, ((T1\*0.4)+(T2\*0.2)+(T3\*0.4)) from gradebook;*

15. In the space below, write the output from the following MySQL statement: (2 points)

*select student\_id from gradebook where T1 > 80;*

```
+-----+
| student_id |
+-----+
| 11011034  |
| 11778843  |
+-----+
```

16. In the space below, write the output from the following MySQL statement: (3 points)

*select student\_id, T1+5 from gradebook where T1 between 70 and 80;*

```
+-----+-----+
| student_id | T1+5 |
+-----+-----+
| 11560795  | 83   |
| 11900023  | 84   |
+-----+-----+
```

17. Identify all three syntax errors in the MySQL command below. (3 points)

*select enrolled\_students.last\_name, enrolled\_students.first\_name, majors\_char\_code  
from enrolled\_students, majors where (enrolled\_students.majorid = majors.id) and  
(majors.char\_code = CSCI");*

Underscore should be a period. →

Underscore missing. →

Missing opening quotation marks. →

18. The data type definition DECIMAL(10, 4): (2 points)

- a.) defines a decimal value between 4 and 10.
- b.) defines a ten digit base four value.
- c.) defines a four digit base ten decimal value.
- d.) defines a ten digit fixed point decimal value with 4 digits after decimal point.
- e.) is an illegal data type definition.

19. In the client/server side model, having a middle layer provides: (circle all that are true) (3 points)

- a.) ability to access more than one database with a single transaction
- b.) ability connect to many different types of data sources
- c.) ability to prioritize requests before they reach the data base
- d.) improved security