**MSDS 7330 File Organization and Database Management**

**Midterm Answer Sheet**

Name: Andrew Abbott

Date: October 22, 2016

1. B
   * I believe (a) to be false (and therefore (d) also) because DDL output creates metadata in the data dictionary (pg 12 of textbook). (b) and (c) are both true statements in regard to DDL. I did find some evidence that Java also supports authorization (<http://docs.oracle.com/javase/7/docs/technotes/guides/security/jaas/JAASRefGuide.html>) so I chose (b).
2. D
   * A list of disadvantages of using a file processing system is on pages 3-6 of the textbook. The list does not include (d), which also is not a disadvantage and is not limited to file processing systems.
3. A
   * In the textbook, on pages 15-16, the design process is described. The first step is to “characterize fully the data needs of the prospective database users.” This has to be done before the schema can be conceptualized. (b) is part of the schema design, which in turn mist be done before the logical and physical design phases (c). Finally the database is created (d).
4. D
   * In the textbook, on pages 15-16, the design process is described. The first step is to (a) “characterize fully the data needs of the prospective database users.” This has to be done before the schema can be conceptualized. (b) is part of the schema design, which in turn mist be done before the logical and physical design phases (c). Finally, the database is created (d).
5. C
   * According to page 6 of the textbook, the three levels of data abstraction are Physical, Logical, and View levels.
     1. Physical level describes how the data are stored.
     2. Logical level describes the data and relationships between the data.
     3. View level simplifies interaction with the system to what is needed or authorized by users.
6. A
   * I eliminated (c), (d), and (e) because object-based models are well suited to object oriented languages, which SQL is not and Semistructured data models are good for languages like XML rather than SQL. (a) and (b) are both well suited for use with SQL but (b) is more involved with the database design than with the language. The relational model is a set of two dimensional tables containing records with attributes. Source: Textbook page 9.
7. E
   * A: According to page 10 of the text, database schema are specified using DDL
   * B and D: Page 11 states that “W specify the storage structure and accss methods used by the database system by a set of statements in a special type of DDL”.
   * C: Also on page 11 the textbook explains that DDL allows consistency constraints to be specified.
8. C
   * A: SELECT AVG(salary)

FROM instructor

WHERE salary < 50000;

* + B: SELECT deptName, AVG(salary)

FROM instructor

GROUP BY deptName;

* + D: SELECT deptName, AVG(salary)

FROM instructor

WHERE salary < 50000

GROUP BY deptName;

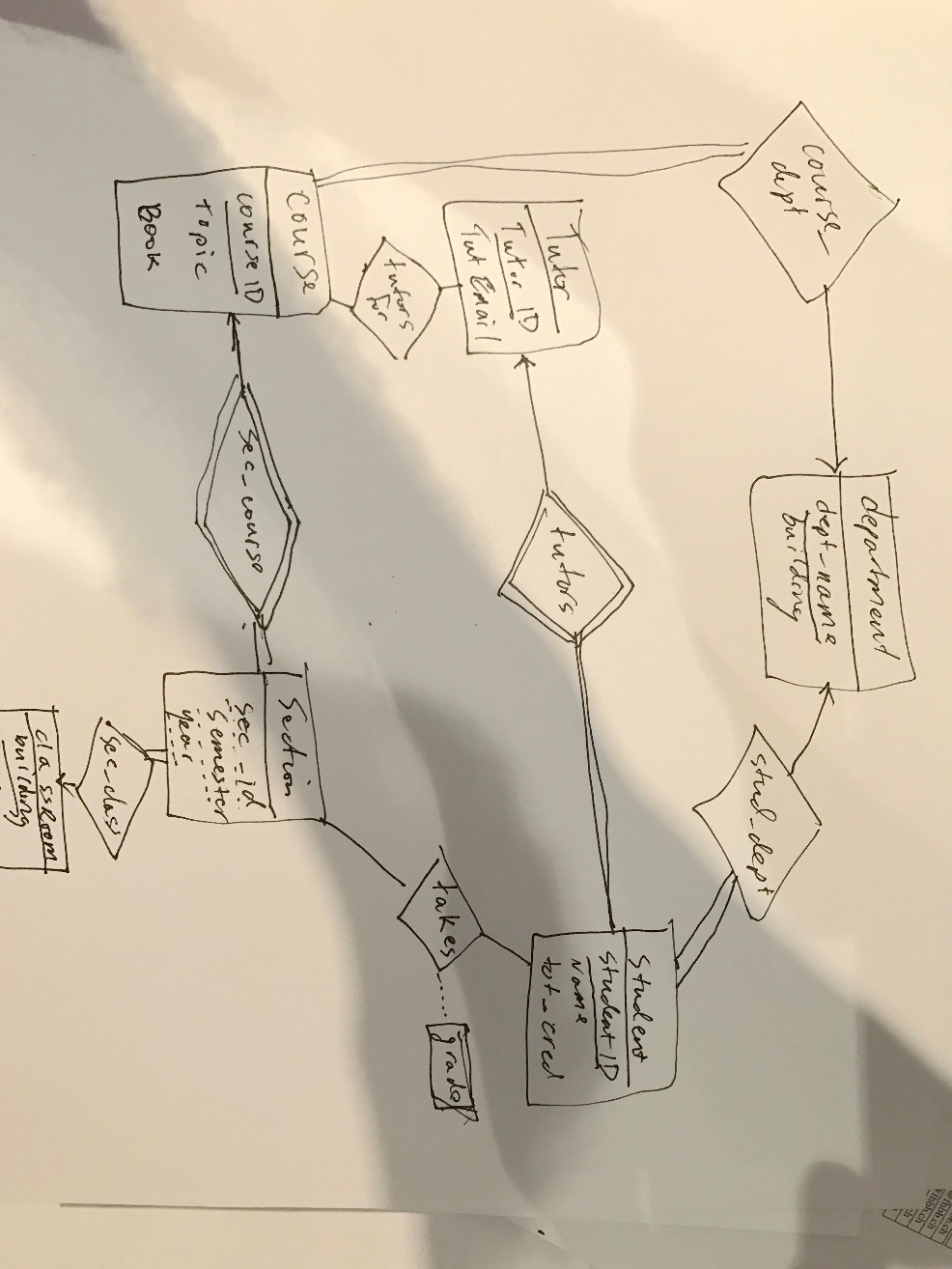
1. B
   * The three tier system separates the business logic of the application from the user by putting it on an application server. The user interfaces with an application client which communicates with the application server. A two tier system is not appropriate for a www applications. Four and five tier applications would have unnecessary tiers. Source; textbook page 25.
2. D
   * Codd described a problem of data inconsistency. Modern database systems allow for consistency constraints.
   * <http://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf>
3. B
   * A primary key is the candidate key chosen as the primary means used to identify a tuple in a relation. Superkeys include all sets of attributes that together uniquely identify tuples and foreign keys are keys in another relation. Source: textbook page 45.
4. C
   * Superkeys are defined in the textbook on page 45 as “a set of one or more attributes that, taken collectively, allow us to identify a tuple in the relation.” Every combination of attributes that are unique to every existing tuple is a superkey.
5. A
   * The primary key is chosen from the candidate keys. Candidate keys are all superkeys that don’t contain subsets that are also superkeys. Textbook page 45.
6. D
   * A foreign key is an attribute in a relation that is the primary key in another relation. This referencing relation must maintain referential integrity with the referenced relation. Any value in the foreign key must also appear in the referenced relation as a primary key. Textbook page 46.
7. B
   * The table name is Student-Tutor which gives me an idea of the purpose of the table. Every combination of studentID and TutorID is unique, so I chose the combination as the primary key.
8. D
   * I chose D because each of A, B, and C should have their own tables with information related to those entities. There should be a student table, a tutor table and a course table with each of the choices as their primary key.
9. I broke the table up by determining which attributes functionally determine which other attributes and within the new tables, if any attributes were dependent on a non-key attribute broke it down further.

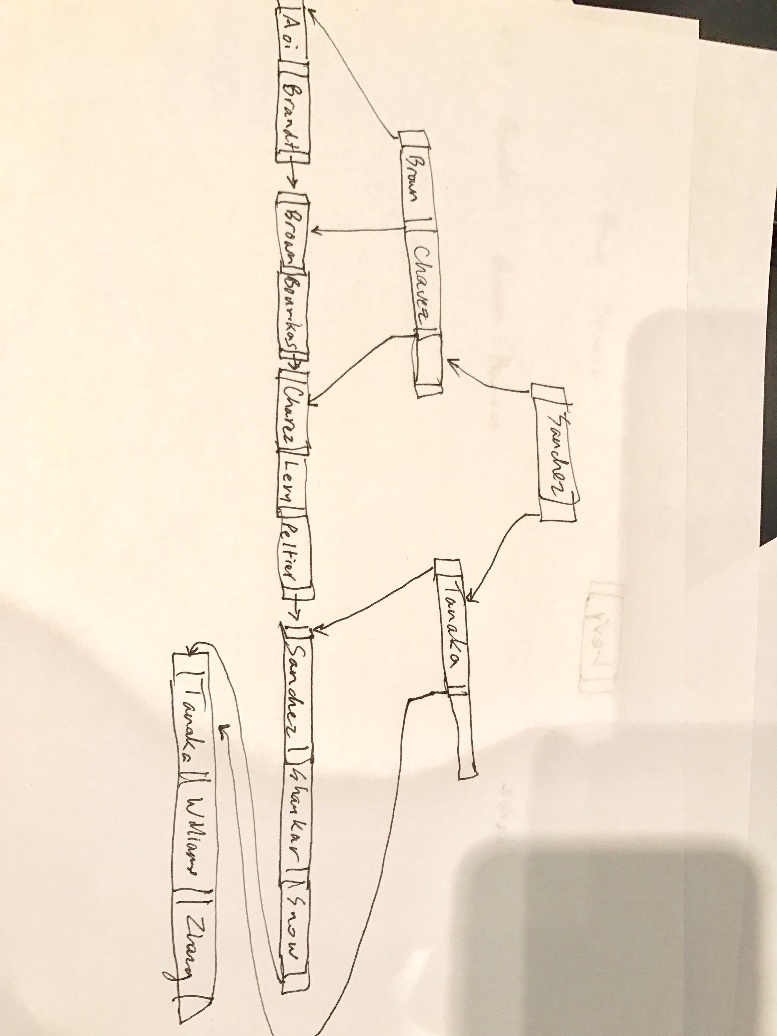
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CourseID | TutorID | Topic | Room | Book | Date |
| U1 | Tut1 | GMT | 629 | Deumlich | 23.02.03 |
| U2 | Tut3 | Gln | 631 | Zehnder | 18.11.02 |
| U5 | Tut3 | PhF | 632 | Dmmlers | 05.05.03 |
| U4 | Tut5 | AVO | 621 | SwissTopo | 04.07.03 |

|  |  |
| --- | --- |
| TutorID | TutEmail |
| Tut1 | [Tut1@fnbb.ch](mailto:Tut1@fnbb.ch) |
| Tut3 | Tut3@fnbb.ch |
| Tut5 | Tut5@fnbb.ch |

|  |  |  |
| --- | --- | --- |
| CourseID | StudentID | Grade |
| U1 | St1 | 4.7 |
| U2 | St1 | 5.1 |
| U1 | St4 | 4.3 |
| U5 | St2 | 4.9 |
| U4 | St2 | 5.0 |

1. My requirements are to represent the university enterprise. Each course has one department. Each student has one department. Each student takes multiple sections that results in a grade, etc.



1. A
   * I chose personName because each person has a unique company, salary and hiredate.
2. C
   * I chose personName, hireDate because actually a person may share a name with another person or may be rehired at the company. (a) could also be appropriate to account for an employee salary chaning.
3. 
4. use university;

select s.name

from student s

left join takes t

on s.ID = t.ID

where t.course\_id like 'CS%'

group by s.name;

|  |
| --- |
| Bourikas |
| Brown |
| Levy |
| Shankar |
| Williams |
| Zhang |

1. Since all courses in the database are on or after Spring 2009, the entire student list is produced.

use university;

select s.ID, s.name

from student s

left join takes t

on s.ID = t.ID

where t.year>=2009

group by s.ID, s.name;

|  |  |
| --- | --- |
| 00128 | Zhang |
| 12345 | Shankar |
| 19991 | Brandt |
| 23121 | Chavez |
| 44553 | Peltier |
| 45678 | Levy |
| 54321 | Williams |
| 55739 | Sanchez |
| 76543 | Brown |
| 76653 | Aoi |
| 98765 | Bourikas |
| 98988 | Tanaka |

1. use university;

select dept\_name as Department, max(salary) as Max\_salary

from(

select d.dept\_name, i.name, i.salary

from department d

left join instructor i

on d.dept\_name = i.dept\_name

) as a

group by dept\_name

|  |  |
| --- | --- |
| Biology | 72000.00 |
| Comp. Sci. | 92000.00 |
| Elec. Eng. | 80000.00 |
| Finance | 90000.00 |
| History | 62000.00 |
| Music | 40000.00 |
| Physics | 95000.00 |

1. use university;

Select min(Max\_salary) as minimum

from(

select dept\_name as Department, max(salary) as Max\_salary

from(

select d.dept\_name, i.name, i.salary

from department d

left join instructor i

on d.dept\_name = i.dept\_name

) as a

group by dept\_name) as b

40000.00