# Quiz 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | hese are the automatically computed results of your exam. Grades for essay questions, and comments from your instructor, are in the "Details" section below. | |  |  | | --- | --- | | Date Taken: | **11/5/2011** | | Time Spent: | **37 min , 16 secs** | | Points Received: | **40 / 40  (100%)** | | | |  |  |  | | --- | --- | --- | | Question Type: | # Of Questions: | # Correct: | | Multiple Choice | 5 | 5 | | Essay | 3 | N/A | | | |

|  |
| --- |
| Grade Details |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | 1. | Question : | TCO A: A database is considered "self-describing" because \_\_\_\_\_. | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: | ORRECT | it contains a description of its own structure | |  |  |  | it contains a listing of all the programs that use it | |  |  |  | it reduces data duplication | |  |  |  | all the users' data is in one place | |  | Instructor Explanation: | Page 12. | | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **2 of 2** | |  | Comments: |  | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | 2. | Question : | TCO A: Business organizations have resisted adopting object-oriented database management systems because \_\_\_\_\_.  A) the cost of purchasing OODBMS packages is prohibitively high B) object-oriented programming uses simplified data structures that fit easily into relational databases C) the cost of converting data from relational databases to OODBMSs is too high D) most large organizations have older applications that are not based on object-oriented programming | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | A | |  |  |  | B | |  |  |  | C | |  |  |  | D | |  |  | ORRECT | Both C and D | |  | Instructor Explanation: | Page 23. | | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **2 of 2** | |  | Comments: |  | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | 3. | Question : | TCO C: A table that meets the definition of a relation is in \_\_\_\_\_. | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | Third Normal Form | |  |  |  | Fourth Normal Form | |  |  |  | Boyce-Codd Normal Form | |  |  |  | Second Normal Form | |  |  | ORRECT | First Normal Form | |  | Instructor Explanation: | Page 101. | | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **2 of 2** | |  | Comments: |  | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | 4. | Question : | TCO C: A tuple is also known as a \_\_\_\_\_. | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | table | |  |  |  | relation | |  |  |  | file | |  |  | ORRECT | row | |  |  |  | field | |  | Instructor Explanation: | Page 98. | | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **2 of 2** | |  | Comments: |  | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | 5. | Question : | TCO C: An attribute is also known as a \_\_\_\_\_. | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | relation | |  |  |  | file | |  |  | ORRECT | field | |  |  |  | table | |  |  |  | row | |  | Instructor Explanation: | Page 99 | | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **2 of 2** | |  | Comments: |  | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | 6. | Question : | TCO A: What are the three types of database design situations? | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | Database constructed from existing data: Team is required to design an database based on the data given in a spreadsheet or other tables. Extracting data from other databases for other use. Design using normalization principles. Database Design for New Systems Development: Required to create a new data model from application requirements and transform data model into database design such as forms and reports. Database Redesign: Where we migrate databases to newer database because of business requirements or process. We can also Integrate two or more databases together because it can optimize process. Also we can Reverse engineer and design new databases using normalization principles and data model transformation | |  | Instructor Explanation: | Database designs can be based on (1) existing data, (2) new systems development, and (3) database redesign. Database designs from existing data may be based on data in spreadsheets, or other data tables, or on data extracted from other databases. New systems development gathers user requirements for needed applications, and designs a database to meet those requirements. Database redesign may be needed to migrate existing databases to a newer DBMS, or to integrate multiple existing databases. | | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **10 of 10** | |  | Comments: | Perfect | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | 7. | Question : | TCO A:  Explain why a small database is not necessarily simpler than a large one. | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | Because the difference between a small database and a large database is really mostly the data that is being entered into it. For example a small shipping company ships 1 million packages a year where a larger company can ship 100 million packages. Even though their quantity are different they stil share the same business model , same kind of data and same kind of logic and levels of complexity. The same amount of complexity applies to both. Thus just because a database is smaller does not make it simpler. | |  | Instructor Explanation: | While small databases vary from large databases, in terms of the amount of data they store, they can still have the same structures and components—types of data, number of tables, and complexity of data relationships—as a large database. A small database is not necessarily a simple database. | | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **10 of 10** | |  | Comments: | Explained exactly correct | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | 8. | Question : | TCO C: What are the advantages and disadvantages of normalization? | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | Lets keep it simple Advantages: First we reduce duplicate data which saves file space and data integrity problems. we also eliminate modification anomalies. Disadvantages: Queries are going to be longer and more complicated. We are gonna need to make joins between tables and even at times to subqueries which I'm not a big fan of. The more joins and subqueries to do, the more work we are putting on the DBMS which can slow the overall application down. Sometimes keeping your DB denormalized is better for business. | |  | Instructor Explanation: | The advantages of normalization are that it (1) eliminates modification anomalies and (2) reduces duplicated data. Reducing duplicated data will eliminate data integrity problems and save file space. The disadvantages of normalization are that it (1) will require application developers to write more complicated SQL statements for multi-table subqueries, and joins and (2) may slow down the applications running against the database. | | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **10 of 10** | |  | Comments: | And, keeping it simple is the best way to go... | | | |

|  |
| --- |
| \* Times are displayed in (GMT-07:00) Mountain Time (US & Canada) |

# Quiz 2

|  |
| --- |
| Grading Summary |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | These are the automatically computed results of your exam. Grades for essay questions, and comments from your instructor, are in the "Details" section below. | |  |  | | --- | --- | | Date Taken: | **11/19/2011** | | Time Spent: | **1 h , 00 secs** | | Points Received: | **45 / 50  (90%)** | | | |  |  |  | | --- | --- | --- | | Question Type: | # Of Questions: | # Correct: | | Multiple Choice | 5 | 5 | | Essay | 4 | N/A | | | |

|  |
| --- |
| Grade Details |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | 1. | Question : | TCO B: Entities of a given type are grouped into an \_\_\_\_\_. | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | entity attribute | |  |  |  | entity relationship | |  |  | ORRECT | entity class | |  |  |  | entity instance | |  |  |  | None of the above | |  | Instructor Explanation: | Page: 145 | | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **2 of 2** | |  | Comments: |  | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | 2. | Question : | TCO B: Which of the following is not a key element of an ER model? | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | Identifiers | |  |  |  | Entities | |  |  |  | Relationships | |  |  |  | Attributes | |  |  | ORRECT | Objects | |  | Instructor Explanation: | Page: 180 | | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **2 of 2** | |  | Comments: |  | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | 3. | Question : | TCO B: You are given an ER diagram with two entities, ORDER and CUSTOMER. If a given customer can place many orders, and a given order can be by one or more customers, which of the following relationships is specified between the two entities? | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | 0:1 | |  |  |  | 1:1 | |  |  |  | 1:N | |  |  | ORRECT | N:M | |  |  |  | N:1 | |  | Instructor Explanation: | Page: 149 | | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **2 of 2** | |  | Comments: |  | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | 4. | Question : | TCO C: Each attribute of an entity becomes a(n) \_\_\_\_\_ of a table. | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | primary key | |  |  | ORRECT | column | |  |  |  | alternate key | |  |  |  | foreign key | |  |  |  | column or foreign key | |  | Instructor Explanation: | Page: 197 | | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **2 of 2** | |  | Comments: |  | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | 5. | Question : | TCO C: Many-to-many relationships are represented by \_\_\_\_\_. | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | an intersection table which has M:N relationships with the two tables | |  |  |  | two tables with a 1:N relationship | |  |  |  | two intersection tables which each have 1:N relationships with the two tables | |  |  |  | two tables with an M:N relationship | |  |  | ORRECT | an intersection table which has 1:N relationships with the two tables | |  | Instructor Explanation: | Page: 200 | | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **2 of 2** | |  | Comments: |  | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | 6. | Question : | TCO C: Explain the pragmatic reason for using surrogate keys. | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | Primary keys are often duplicated as foreign keys to represent relationships. When the primary key contains a lengthy text field, this creates a large amount of duplicated data that must be frequently manipulated, thereby degrading database performance. Therefore, we often use a surrogate key that is generated by the system and is relatively small and easy to manipulate. | |  | Instructor Explanation: | Primary keys are commonly included in indexes and are used to identify records to be retrieved by users. The ideal primary key is short, numeric, and fixed. When the primary key contains a lengthy text field, this creates a large amount of duplicated data that must be frequently manipulated. For these reasons, it is often practical to use a surrogate key that is generated by the system and is relatively small and easy to manipulate.  Page: 194 | | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **10 of 10** | |  | Comments: | Good answer. | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | 7. | Question : | TCO C: What are MUST NOT and MUST COVER constraints? | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | MUST COVER constraint: is the binary relationship that indicates all the combinations that must apear in the ternary relationship. For example, let say you need to take your car to a shop to get repaired. for for the specific repair they MUST do certain task to repair the problem. AUTO:REPAIR:TASK. A given REPAIR contains a number of tasks which all MUST be performed for the repair to be successful. In this example wen AUTO has a given REPAIR, then all the TASKS for the REPAIR must appear as rows in that relation. MUST NOT: Is the opposite of MUST COVER. The binary relationship which indicates the combinations that are absolutely not allowed to occur in a ternary relationship. Example, a patient goes to doctor and the doctor perscribes them something but he needs to make sure that the patient is not allergic to the drug. DRUG:PATIENT:ALLERGY Given the PATIENT allergies, the drug must not contain any ingredients that the patient is allergic to. | |  | Instructor Explanation: | Both MUST NOT and MUST COVER constraints occur when working with three tables in a ternary relationship. When a MUST NOT constraint occurs, it indicates that one binary relationship of two of the tables contains combinations of values that are not allowed to occur in the ternary relationship. When a MUST COVER constraint occurs, it indicates that one binary relationship of two of the tables contains combinations of values that must all appear in the ternary relationship.  Page: 210–212 | | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **10 of 10** | |  | Comments: | Excellent | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | 8. | Question : | TCO B: What is meant by the cardinality of a relationship? Your answer needs to be thorough and specific. Be sure to include the two types of cardinality and also include the four possible primary relationship cardinalities. | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | It basically means the number of elements allowed on each side of the relationship. Maximum Cardinality: is the maximum number of entity instances that can participate in a relationship instance Minimum Cardinality: is the minimum number of entity instances that must participate in a relationship In Maximum Cardinality we have: 1:1 (one-to-one) - where an entity of one type is related to at most one entity of the other type. ex. 1 person has at most 1 social security number 1:M (one-to-many) - where an entity of one type is related to many instances of the other type. ex. 1 person can have many cars. N:M (many-to-many) - where an entity of parent instance can be assoicated with many child instances and one child entity instance can be associated to many parent entity instances. ex. 1 employe can have many skills. and the same skill and the same skill can be done by many employees. M:M (mandatory | |  | Instructor Explanation: | In the ER model, there are two types of cardinality; the maximum cardinality and the minimum cardinality. The maximum cardinality is the maximum number of entity instances that can participate in a relationship instance. There are three types of maximum cardinality one–to–one (11), one–to–many (1N), and many–to–many (NM). The minimum cardinality is the minimum number of entity instances that must participate in a relationship instance. This number is generally zero (0), or one (1). If the number is zero, then participation by that entity instance is optional(O), it does not have to be in a relationship with an instance of the other entity. If the number is one, then participation by that entity instance is mandatory, (M)ðDit must participate in a relationship with an instance of the other entity. In a binary relationship there are four (4) possible sets of minimum cardinalities (O-O), (O-M), (M-O), and (M-M). | | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **7 of 10** | |  | Comments: | You did an excellent job with the Maximum Cardinality possibilities, but you didn't cover the 4 possibilities for Minimum cardinalities. | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | 9. | Question : | TCO B: What are supertype and subtype entities? | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | Supertype: Is an entity that logically contains subtypes. For example, EMPLOYEE is a suppertype of ENGINEER, SALESPERSON, DIRECTOR Subtype: Is an entity that that is a subcategory of a suppertype. for example, DIRECTOR is a subtype of EMPLOYEE. | |  | Instructor Explanation: | When instances of an entity can be categorized into different types, with each type sharing some common characteristics, while having certain characteristics that are unique to each type, the common attributes can be separated from the unique ones. This is represented in ER diagrams through the use of supertype and subtype entities. The supertype entity class contains the attributes that are common to all of the subtypes. The subtype entity classes represent the different types, or categories, of the supertype. Only the attributes that are unique to a given subtype tend to be listed as the attributes of that subtype. Each subtype has a relationship with the supertype. All subtypes should have the same identifier as the supertype since they represent different perspectives of the same thing. | | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **8 of 10** | |  | Comments: | They are more than "logically containing subtypes" | | | |

# Quiz 3

|  |
| --- |
| Grading Summary |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | These are the automatically computed results of your exam. Grades for essay questions, and comments from your instructor, are in the "Details" section below. | |  |  | | --- | --- | | Date Taken: | **12/6/2011** | | Time Spent: | **34 min , 02 secs** | | Points Received: | **46 / 50  (92%)** | | | |  |  |  | | --- | --- | --- | | Question Type: | # Of Questions: | # Correct: | | Multiple Choice | 5 | 4 | | Essay | 4 | N/A | | | |

|  |
| --- |
| Grade Details |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | 1. | Question : | TCO D: When one SQL query is embedded in the WHERE clause of another SQL query, this is referred to as a \_\_\_\_\_. | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | WHERE query | |  |  |  | set query | |  |  |  | join | |  |  |  | subset | |  |  | ORRECT | subquery | |  | Instructor Explanation: | Pages: 70–71 | | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **2 of 2** | |  | Comments: |  | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | 2. | Question : | TCO D: In an SQL query, which SQL keyword actually creates the query? | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | FROM | |  |  |  | WHERE | |  |  | ORRECT | SELECT | |  |  |  | SET | |  |  |  | EXISTS | |  | Instructor Explanation: | Page: 36 | | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **2 of 2** | |  | Comments: |  | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | 3. | Question : | TCO D: If the values in an SQL view are changeable through the view itself, which SQL statement is used to change the values? | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | INSERT | |  |  |  | SELECT | |  |  |  | DELETE | |  |  | ORRECT | UPDATE | |  |  |  | CREATE | |  | Instructor Explanation: | Page: 259 | | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **2 of 2** | |  | Comments: |  | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | 4. | Question : | TCO D: Which SQL keyword is used to add one or more rows of data to a table? | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | SET | |  |  |  | SELECT | |  |  |  | DELETE | |  |  | ORRECT | INSERT | |  |  |  | UPDATE | |  | Instructor Explanation: | Pages: 252–253 | | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **2 of 2** | |  | Comments: |  | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | 5. | Question : | TCO D: To add a NOT NULL column to a table, we \_\_\_\_\_. | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | use the REVISE TABLE command | |  |  | ORRECT | create a new NULL column, insert data values into every row, and change the NULL constraint to NOT NULL | |  |  | NCORRECT | use the ALTER TABLE command | |  |  |  | use the MODIFY TABLE command | |  | Instructor Explanation: | Page: 307 | | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **0 of 2** | |  | Comments: |  | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | 6. | Question : | TCO D: Explain the essential format of the CREATE TABLE statement. | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | The SQL CREATE TABLE statement is the SQL command that adds a new table to an SQL database. The table has column names, each column has a datatype and a definition, and constraints. table constraints are primary keys, foreign key, not null, null, unique, check CREATE TABLE NewTable ( attribute datatype definition table constraints ) | |  | Instructor Explanation: | The essential format for the CREATE TABLE statement is: CREATE TABLE tablename (column–description, column–description, column–description, . . . optional table constraints); "Tablename" is the name that will be given to the newly created table. "Column–description" is a three–part description of each column to appear in the table. This description includes the name of the column, the column's data type, and an optional column constraint—either Primary Key, Null, or Not Null, in that order. The CONSTRAINT phrase can be used to set optional primary key, foreign key, and referential integrity constraints for the table. All SQL statements must end with a semicolon (;). Page: 238–239 | | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **10 of 10** | |  | Comments: | Well put... | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | 7. | Question : | TCO D: Briefly describe subqueries and joins. Explain when each is not an acceptable alternative for the other. | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | subqueries are queries inside other queries. We usually use subqueries when we are comparing the column of one table with a result set of another query. ex. SELECT COUNT(\*) AS DeptCount FROM (SELECT DISTINCT Department FROM SKU\_DATA) AS DEPT; You can see Joins as the result of a Venn Diagram and each table is a circle. When we do joins we can take choose to display any of the columns of the two tables. Left joins consist of the intersection and the left most of the two tables. and vice versa for right joins. Outer Joins are everything but the intersection and Inner joins are only the intersection. When possible doing joins can be better as subqueries are costly. | |  | Instructor Explanation: | Subqueries and joins are both methods for retrieving data from multiple tables. Subqueries involve nesting one SELECT statement within another. The nested SELECT is used as part of a condition in the WHERE clause of the first SELECT statement. The nested SELECT statement can return a set of records from one table, which are then used in a logical operator within the parent SELECT query. A join combines records from each table into concatenated records containing the fields of both tables. The records are concatenated based on matching values in similar columns in the two tables. Subqueries cannot be used in situations where the results to be displayed include attributes from more than one table. Joins cannot be used as an alternative to a correlated subquery. Page: 70–77 | | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **10 of 10** | |  | Comments: | Yes. The key is that joins you can display ANY column from any of the two tables. | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | 8. | Question : | TCO D: What are SQL triggers, and how are they used? | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | Triggers are a kind of stored procedure that are invoked whenever an attempt is made to modify the data in the table it protects. Triggers are used to enforce data integrity and business rules such as automatically updating summary data. It allows to perform cascading delete or update operations. If constraints exist on the trigger table,they are checked prior to the trigger execution. | |  | Instructor Explanation: | An SQL trigger is a stored program that is attached to a table or view. The trigger is invoked by the DBMS whenever an insert, update, or delete request is made on the table, or view with the trigger. There are three commonly used triggers: BEFORE, INSTEAD OF, and AFTER (MS SQL server does not support BEFORE). This creates a set of nine possible trigger types: BEFORE + [INSERT or UPDATE or DELETE], INSTEAD OF + [INSERT or UPDATE or DELETE], and AFTER + [INSERT or UPDATE or DELETE]. Triggers are used (among other things) for (1) providing default values, (2) validity checking, (3) updating views, and (4) enforcing referential integrity actions. Page: 275–76, Fig 7–25 | | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **10 of 10** | |  | Comments: | Excellent | | |  |  |  |  |  | | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | 9. | Question : | TCO D: What is the process and what are the considerations when changing a table name? | | | |  |  |  |  | | --- | --- | --- | --- | |  | Student Answer: |  | well in mysql it is quite simple you just use the Rename keyword. RENAME TABLE tbl\_name TO new\_tbl\_name. However, before we even do this we have to first, duplicate the original table. Then we rename our duplicated table. 3rd we start renaming the foreign key constraints to our new table. we remove the constraints from our old table. we drop the old table. | |  | Instructor Explanation: | To change a table name, we re–create the table with the new name, copy the data to the new table, and finally drop the old table. Although the process is simple, there are several considerations. First is the problem of a surrogate key, for the existing values must be maintained. This is solved by creating the key as regular field, copying the date, and then converting the column to a surrogate key. Second is the problem of restructuring constrains to make sure they now apply to the new table. Finally any stored procedures, triggers, and other application code that applied to the old table, must be rewritten to apply to the new table. Page: 305–306 | | |  | | | | | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | |  | Points Received: | **8 of 10** | |  | Comments: | I'm going to assume that "duplicate" and "rename" the original table is create and copy. But we have to remove constraints from other tables and add new constraints (if needed). Also, you have to consider stored procedures and triggers. | | | |

|  |
| --- |
| \* Times are displayed in (GMT-07:00) Mountain Time (US & Canada) |
| |  |  |  | | --- | --- | --- | | **Week, TCOs and Topics** | **Readings/Class Preparation** | **Activities/Assignments** | | **Week 1 TCO A**  Introduction to Database Processing | Chapter 1: Introduction   Lecture Tutorial: Getting Started with Access 2010 | Discussion (2 Graded Topics) | | **Week 2 TCO C**  Normalization | Chapter 3: The Relational Model and Normalization   Chapter 4: Database Design Using Normalization   Chapter 5: Data Modeling with the Entity-Relationship Model | Course Project  Discussion (2 Graded Topics)  Quiz | | **Week 3 TCO B**  Database Modeling | Chapter 5: Data Modeling with the Entity-Relationship Model | Discussion (2 Graded Topics) | | **Week 4 TCO C**  Transforming Data Models | Chapter 6: Transforming Data Models in Database Designs | Course Project  Discussion (2 Graded Topics)  Quiz | | **Week 5 TCO D**  Structured Query Language (SQL) | Chapter 2: Introduction to Structured Query Language | Course Project  Discussion (2 Graded Topics) | | **Week 6 TCO D**  Relational Redesign | Chapter 7: SQL for Database Construction and Application Processing   Chapter 8: Database Redesign | Discussion (2 Graded Topics)  Quiz | | **Week 7 TCO E, F, G**  Multiuser Databases, Data Warehouses and Database Administration | Chapter 9: Managing Multiuser Databases   Chapter 13: Database Processing for Business Intelligence Systems | Course Project  Discussion (2 Graded Topics) | | **Week 8 All TCOs**  All Topics | **Final Exam** | | |