

Project 2

Chapter10:

1 The main database transaction properties are atomicity, consistency, durability, isolation, and _____.

- a. serializability
 - b. capability
 - c. concurrency
 - d. transitivity
-

2 In database terms, a(n) _____ is any action that reads from and/or writes to a database.

- a. lock
 - b. session
 - c. transaction
 - d. interaction
-

3 A _____ database state is one in which all data integrity constraints are satisfied.

- a. fixed
 - b. secure
 - c. complete
 - d. consistent
-

4 If any of the SQL statements in a transaction fail, the entire transaction is _____ to the original database state that existed before the transaction started.

- a. abandoned
 - b. completed
 - c. rolled back
 - d. diverted
-

5 All transactions are controlled and executed by the DBMS (subject to some limitations) to guarantee database _____.

- a. uniqueness
 - b. integrity
 - c. consistency
 - d. design
-

6 What is a database request?

- a. A series of SQL statements that implement all of the business rules in an application.
 - b. One or more SQL statements that implement one business rule.
 - c. A single SQL statement in an application program or transaction.
 - d. A transaction.
-

7 Each database request generates several _____.

- a. input/output (I/O) operations
- b. transactions
- c. business processes
- d. SQL statements

8 The _____ statement is used to end a successful transaction.

- a. COMMIT
 - b. DONE
 - c. END
 - d. QUIT
-

9 Which DBMS does not support transaction management by default?

- a. MS Access
- b. Oracle
- c. DB2
- d. SQL Server

(Also, MySQL default tables, MyISAM, does not support transactions.)

10 _____ requires that all operations of a transaction be completed; if not, the transaction is aborted.

- a. Atomicity
 - b. Durability
 - c. Consistency
 - d. Isolation
-

11 _____ means that the data used during the execution of a transaction cannot be used by a second transaction until the first one is completed.

- a. Isolation
 - b. Atomicity
 - c. Durability
 - d. Consistency
-

12 _____ ensures that once transaction changes are done, they cannot be undone or lost, even in the event of a system failure.

- a. Consistency
 - b. Isolation
 - c. Atomicity
 - d. Durability
-

13 Which property is especially important in a multiuser database?

- a. Consistency
 - b. Atomicity
 - c. Serializability
 - d. Durability
-

14 If several concurrent transactions are executed over the same data set and the second transaction updates the database before the first transaction is finished, the _____ property is violated and the database is no longer consistent.

- a. atomicity
- b. consistency
- c. durability
- d. isolation

15 When a(n) _____ statement is reached, all changes are permanently recorded within the database.

- a. DONE
- b. EXIT
- c. COMMIT
- d. ROLLBACK

16 When the end of a program is reached, all changes are permanently recorded within the database; this action is equivalent to issuing the _____ statement.

- a. EXIT
- b. BYE
- c. COMMIT
- d. ROLLBACK

17 When a program is abnormally terminated, the equivalent of a(n) _____ command occurs.

- a. COMMIT
- b. ROLLBACK
- c. QUIT
- d. EXIT

18 According to the ANSI SQL standard, when does a transaction begin?

- a. with the BEGIN TRANSACTION command
- b. when the first SQL statement is encountered
- c. with use of the START command
- d. with the BEGIN command

19 A DBMS uses a transaction _____ to keep track of all transactions that update the database.

- a. block
- b. log
- c. statement
- d. table

20 Part of the data stored in the transaction log is a(n) _____ to the previous and next transaction log entries for the same transaction.

- a. rowid
- b. pointer
- c. id
- d. link

Deliverables

MS SQL Server

Design Description and Business Rules:

For the most part, database designs involve binary relationships: that is, a relationship between two entities (tables). Sometimes, however, more than two entities are represented in the participation constraints—based upon, of course, the business rules. When putting together the logical and physical designs, generally, an associative entity is used to bridge the “n-ary” (more than two) relationship participations. This particular database design reflects a patient treatment. That is, a patient, may receive a treatment, as well as maybe a medication, from a physician.

The following data must be kept: a doctor’s specialty, name, address, phone, fax, e-mail, url, and any pertinent notes. Likewise, similar patient data must be kept—though, fax or url data are not necessary. A treatment should include its description, price, and any further notes. A medication should include its name, price, shelf-life, and any additional notes. Prescription should record date, dosage, number of refills, and notes. Finally, patient treatments should track the patient, physician, treatment, date, start time, end time, results, and notes. Also, include the following business rules:

- A physician will prescribe one or more treatments.
- A treatment can be prescribed by one or more physicians.
- A patient can receive one or more treatments.
- A treatment can be prescribed to one or more patients (implicitly, by more than one doctor).
- Each prescribed treatment on a specific date, can be prescribed by only one physician to one patient.
- A patient can have many medications. Many medications can be given to many patients.
- A prescription can be administered in one or more treatments.
- A treatment can include one or more prescriptions.

T-SQL and Diagramming Requirements

1. Bb > Notes > Log into SQL Server

Note: Do ***NOT*** have a diagram window open when attempting to modify your database!

2. Use ***only*** T-SQL to create tables, relationships, constraints, and data from the above business rules. **Tables *MUST* be populated on CCI’s MS SQL Server to receive credit.**

3. **Notes: *ALL* tables *must* include the following checks and defaults.**

Updated zip/phone constraints use regular expressions.

- state: default = FL
- ***all*** currency values > 0
- ***all*** enum values use w/in “IN” clause
- zip: require entries in zip column to be **9 digits (no fewer/no more)**

Example: ([pat_zip] like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')

- phone num: require entries in phone column to be **10 digits (no fewer/no more)**

Example: ([pat_phone] like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')

***ALL* FKs: *Must* require ON DELETE NO ACTION, ON UPDATE CASCADE, except administration_lu, which will require one of its FKs to use ON DELETE NO ACTION, ON UPDATE NO ACTION (we will discuss in class why!)**

(When changing to “text” view be sure to use databasename again!)

4. Include **five *unique* records** for each table

5. **Copy and paste query result sets with headers for *ALL* tables (select * from dbo.tablename;).**

Below: Format ***all*** dollar amounts to two places, including dollar sign(\$). Use **alias names** for formatted currency values.

- a. Using **JOIN ON**, create a **transaction** that lists all patients' first, last names, any patients' notes, as well as all of their medication names, prices, shelf lives, prescription dosages and number of refills., order by medicine price in descending order.
 - b. Using **old-style join**, create a **view** (**dbo.v_physician_patient_treatments**) that lists all physicians' first, last names, treatment names, treatment prices, treatment results, as well as start and end dates, order by treatment price in descending order.
 - c. Create a **stored procedure** (**AddRecord**) that adds the following record to the patient treatment table, **and** that uses the view above **after** the added record:
patient id 5, physician id 5, treatment id 5, 4/23/13, 11am, 12:30p, released, ok
 - d. Create a **transaction** that removes ***only*** the fifth record in the administration lookup table—
remember: it is a composite key!
 - e. Create a **stored procedure** (**UpdateRecord**) to update the patient's last name to "Vanderbilt" whose id is 3.
 - f. Use an **alter statement** to create a "**prognosis**" attribute in the patient treatment table capable of storing 255 alpha-numeric characters, just above the "notes" attribute. The attribute value can be null.
6. As per **A5**, create a **database diagram**, save as .png file, and upload it to Bb.

*****Extra Credit***:**

Create a stored procedure (**AddShowRecords**) that adds a patient record when called, **then** displays all treatments (i.e., descriptions) associated with patients and their doctors, include their names, start/end times, sorted by the most recent date.

Helper videos:

1. http://qcitr.com/vids/LIS3784_P2_Tables_and_Data.mp4
2. http://qcitr.com/vids/LIS3784_P2.mp4