Teaching Case

Remote Services, Inc.

Steven A. Morris
Department of Computer Information Systems
Middle Tennessee State University
Murfreesboro, TN 37129
Email: smorris@mtsu.edu

ABSTRACT

The Remote Services, Inc. (RSI) case is designed as an extensible, database design and implementation project. The case is designed in two primary components: design and implementation. The design component of the case allows students to evaluate a scenario that is similar to a real-world business situation and create an appropriate design strategy. The implementation component of the case provides the students with a related scenario that gives them a completed design to be implemented. Implementation tasks include the creation of tables, loading data, creating views, creating triggers, and coding stored procedures. This allows the instructor to easily separate the design and implementation activities so that students that struggle with the design aspect of the project can start with a "clean slate" for the implementation activities, thus giving the instructor a better chance to assess the students' implementation-oriented skills. The case can be used as a group project exercise. It can also be easily extended to include additional human resources and accounting functions for a more complex case.

Keywords: Database Design, Database Implementation, Teaching Case, SQL, Entity Relationship Diagramming

1. CASE SUMMARY

Remote Services, Inc. (RSI) is a computer and office equipment repair company in the tri-city area. RSI has been in business for about five years providing telephone technical support for most name-brand computers, printers, scanners, fax machines, and copiers. RSI has been very successful with its telephone support program, and has grown to the point that it now employs over 20 full-time support technicians. The company provides telephone support that can assist callers in performing most repairs on standard computer and office equipment without the delays and costs associated with having a technician dispatched to perform on-site repairs.

In part 1 of the case, Pat Sherman, the owner and CEO of RSI, has become concerned about the ability to accurately track contracts, customers, and support calls. She described the problem as follows:

> *"Our biggest problem is that we don't know what we have done. When RSI first started, it was fairly easy to keep up with the different companies and the contracts that I had with them, but the company has succeed beyond my expectations. Our focus on being customer-driven and responding to the specific needs of each company has been well received. Gone are the days when one company meant one contract and one client. Now we have multiple contracts with several large companies, and even smaller businesses are registering multiple clients on each contract. Our current system for keeping track of all the clients on all the different contracts and the work we do for them, just isn't keeping up. If RSI is going to continue to grow, we have to have a better handle on what we are doing and who we are doing it for."*

Sydney Wallace was one of the first technicians hired by the fledgling RSI. In addition to strong technical skills, Sydney has proven to have a good head for business and has been promoted to the newly created position of Chief Operating Officer. One of his highest priorities is to maintain RSI's reputation for quick responses to client needs. A reputation that Sydney feels is in jeopardy.

> *"Fast service has been one of the cornerstones of RSI's business model from the beginning. When a client has a problem, they want it fixed immediately – not in an hour. A large part of RSI's growth is attributable to our quick response to client needs. Our rapid growth, however, is making that more difficult and customers are starting to notice. During peak call times, our service technicians are being overwhelmed. The problem is that we struggle with balancing our staffing loads. Our current system doesn't provide any useful information on call volumes, call durations, or the number of technicians that have to participate in the solving of a problem. Without this type of information, we are staffing by trial-and-error."*

You work for a consulting company that has been hired to help design an appropriate database structure to record information to support RSI's business activities. You are working as one part of a small group that has been assigned to work with RSI. Your primary responsibility is to create the design specification for the database structure.

In part 2 of the case, the task expands to include on-site repair services. Pat has decided that now is the time to expand into on-site repairs. She explains the decision as the next logical step in the growth of the company.

> *"RSI has an incredible resource in the knowledge base that our technicians represent. Leveraging that resource to get the most out of it only makes sense. There are many problems that clients report that are difficult, if not impossible, for untrained individuals to complete even with one of our technicians talking them through it. However, we have the skilled personnel to complete these tasks in a timely manner. Providing this service to our clients allows us to continue to provide the most complete solutions to their needs."*

This significantly increases the size of the database development activities that your company will be performing. As a result, new personnel have been added to the task and your role is changing. Other members of the team are taking over responsibility for adding these new requirements to the database design, and still others are working on creating the front-end applications to interact with the database. Your primary responsibility is now to assist with the implementation of the database design. You will be given a design that needs to be implemented and specifications for associated triggers and stored procedures that must be created. Other members of your team will expect you to have these objects created so that the application programs can interface with them.

For part one, based on the description of operations, it will be necessary for you to create an entity-relationship diagram (ERD) data model of the data requirements. This model should be appropriately normalized. Any decision to de-normalize the model should be explained and appropriately justified. For part two, remember that you are part of a larger development team. While you are implementing aspects of the database structure, application programmers are creating the front-end applications. Therefore, in the implementation tasks it is important that table names, attribute names, stored procedure names and view names be implemented exactly as specified since those are the names the application programmers will be using in their code to interact with the database structures you are building.

2. CASE TEXT

2.1 Part 1 – Design Task

Based on interviews with Pat Sherman, and a number of employees at various levels of the organization, the following requirements have been culled. Remote Services, Inc. (RSI) only provides services to companies that have contracted for service. Each service contract has a contract number, beginning date, ending date, annuity amount, problem fee, contract status (open, pending, closed) and maximum clients. Most service contracts are for a 2-year period, although other lengths can be negotiated. The annuity amount is the annual payment that a company makes to maintain the contract. The problem fee is an additional charge assessed each time a company contacts RSI for help with a problem. (If a company expects to use RSI frequently, they may negotiate a higher annuity amount and a lower problem fee. If a company expects to use RSI infrequently, they may negotiate a low annuity amount and a higher problem fee.) The "maximum clients" is the limit on the number of individuals that can register a problem with RSI on that contract. Each contract is with only one company. Over the years, a company may have had several contracts with RSI, all of which need to be represented in the database. All contracts must be associated with a company. A company must have a contract. For each company, the company name, billing address (street, city, state, zip), and contract liaison person name must be stored.

When a company has a contract with RSI, the individuals within the company that wish to use RSI's services must register with RSI. (As described above, the "maximum clients" attribute is the maximum number of individuals from the company that can register with RSI.) For an individual to register with RSI, they must provide a valid contract number. They are assigned a client number, and their name (first and last), title, telephone number, and email address are stored. A client can register using many contracts, and a contract can have many clients registered for it. All clients must be registered for a contract. New contracts may not have any registered clients yet. When a client registers for a contract, the date of registration is also recorded.

Clients call to request assistance with a problem. If this is the first call about a new problem, then the problem is logged and the call is logged. Each new problem is assigned a problem number. For each problem, a description of the problem, the date that it is first reported, its status (open, callback, or closed), fee charged, and the date it is resolved are recorded. It is important to track which client first reported a problem. Further, since a client can be associated with multiple contracts, the contract that the problem is being handled through must be tracked. Any given client can be the first person to report a problem, although some clients can be in the system without having ever been the first person to report a problem. All problems must have been reported by a client. Since the system is supposed to track which client *originally* reported the problem, each problem can only be originally reported by one client. All problems are initially given a status of "open" when it is created. Once the client indicates that the problem is resolved, the date of resolution is recorded, and the problem status is changed to "closed". With some problems (especially intermittent problems that only show up periodically), it is not possible to immediately know if the technician's recommendation has resolved the problem. In this case, the problem status is set to "callback" and, on the next business day, a technician will call the client that originally reported the problem to check on the status of the problem.

All calls for assistance are logged in the system. This includes both the original call to report a new problem (described above), and follow-up calls about a previously

reported problem. Each call is assigned a call ID number; and the date and time of the call, call description, and the call status (X, XX, XXX – described below) are recorded. Additionally, the system needs to track which RSI technician(s) handle the call. Most calls are made by registered clients. If a call is made by an individual that is not a registered client, then the individual must register before they can receive assistance with a problem. If the individual cannot be registered (either because they do not have a valid contract number, or the contract has already exceeded the "maximum clients" for that contract), then the call is still logged but it is not associated with any problem or client. Even in this case, the technician handling the call must be recorded. RSI wishes to log these calls so that complete data on call volume can be maintained and used by management in making staffing decisions.

Each technician is identified by a unique employee number. Additionally, the system stores the technician's name (first and last), date of hire, direct telephone number, telephone extension, and level (1, 2, or 3). The technician level indicates a level of expertise in computer and office equipment troubleshooting. Most phone calls are answered by a level 1 technician. These technicians have a basic level of expertise and can handle most of the common problems that clients report. If a problem can not be resolved by the level 1 technician, a phone call can be escalated to a level 2 technician. Level 2 technicians have much greater expertise in dealing with problems and can handle most of the problems that the level 1 technicians cannot. If the level 2 technician cannot resolve a problem, it can be escalated again to a level 3 technician. RSI has very few level 3 technicians.

While RSI strives to resolve each problem in a single phone call, often a single problem will result in many phone calls. Since problems are only reported through phone calls, all problems will be associated with at least one call. Each call is in reference to only one problem. All calls start with a call status of "X," indicating that it is a normal assistance request. It is possible for a level 1 technician to transfer a call to other level 1 technicians. If a call is escalated to a level 2 technician, the call status is changed to "XX". A level 2 technician can transfer a call to other level 2 technicians. If a call is escalated to a level 3 technician, the status is changed to "XXX". It is possible for a level 3 technician to transfer a call to other level 3 technicians. Each technician that participates in a telephone call, even briefly, needs to be recorded and needs to be able to record a brief description of their efforts to resolve the problem. Additionally, the order in which the different technicians spoke to the client during the phone call should be recorded. Clearly, a telephone call must be handled by at least one technician, and can potentially be handled by many technicians. A technician can handle many telephone calls, although new technicians may not have handled any telephone calls at first.

2.2 Part 2 – Implementation Task

The database should track companies, contracts, clients, problems, appointments, consultants, and visits (see Figure 1). A company has a company identifier (unique number), company name, billing street, billing city, billing state, billing zip, and liaison name. A contract has a contract number, begin date, end date, annuity amount, problem fee, status, and maximum clients. A client has a first name, last name, title, phone number, and email address. When a client registers for service under a contract, the registration date must be recorded. A problem has a problem number, problem description, open date (the date the problem was first reported), a status, fee, and close date (the date the problem was resolved). A problem must be associated with a client (who first reported it) and a contract (under which it is being resolved). A consultant has an employee number, first name, last name, and a cell phone number. An appointment has an appointment number, date, time category, address (street, city, state, and zip) and status. An appointment may yield several different visits. A time category has a time category number and a time category description. A visit has a visit number, date, and visit comments. The first implementation task is to create the tables using the following requirements.

2.2.1 Create tables.

Write the SQL commands necessary to create the following tables (figure 2). All tables and attributes must be created as described in below. The data types given below are broad descriptions of the kind of data that must be stored. It will be necessary to determine the correct data type within the DBMS to implement that kind of data. Be certain that all specified attribute constraints are enforced. Note that due to foreign key constraints, the order in which the tables are created and populated with data is significant.

2.2.2 Insert Data

Write the SQL commands to insert the following data (figure 3) into the tables created above. Note that due to foreign key constraints, the order in which the tables are populated with data is significant.

2.2.3 Update the data.

Write the SQL commands to perform the following updates to the data that has been loaded. Be certain to commit the changes when finished.

1. Update the title for client number 5 to read 'Warehouse Manager'.
2. For contract number 8, set the contract end date to 10/31/2008.

2.2.4 Alter tables.

Write the SQL commands to enact the following changes to the structure of the database tables.

1. Alter the Contract table to use a CHECK table constraint to ensure that the End date is greater than the Begin date.
2. Alter the Contract table to use a CHECK constraint to ensure that the Contract Status is restricted to the domain "OPEN", "PENDING", "CLOSED".

2.2.5 Create triggers.

Write the code to create the following triggers within the database. Note that if updates are made to the data in the tables in order to test the functioning of the triggers, these updates should be rolled back.

1. Create a row-level trigger named TRG_CONTRACT to run before insert or update of the contract begin date of

the Contract table. If contract status is 'OPEN' and the contract end date is null, then the trigger should automatically set the contract end date to be 730 days after the begin date.

2. Create a row-level trigger named TRG_PROB_FEE to run before insert or update of the problem fee on the Problem table. If the user performs an insert or update that would result in a null value for prob_fee in the Problem table, then get the problem fee from the associated contract in the Contract table.
3. Create a trigger named TRG_VISIT to run before insert on the Visit table. The user will always provide an appointment number for the visit being inserted. However, when entering a new record in the VISIT table, if the user does not provide a Consultant employee number, get the consultant employee number from the corresponding appointment. Also, if the user does not provide a Client number, then get the client number that is associated with the problem that is associated with the appointment.
4. Create a trigger named TRG_APP_LOCATION to run before insert on the Appointment table. If the user does not provide a complete address (street, city, state, and zip) for an appointment, then, based on the problem number supplied by the user, get the company's address (street, city, state, zip) from the company table to use as the appointment address (street, city, state, zip). Note, the trigger should only provide the address from the company table if the user does not provide any part of an address for the appointment.

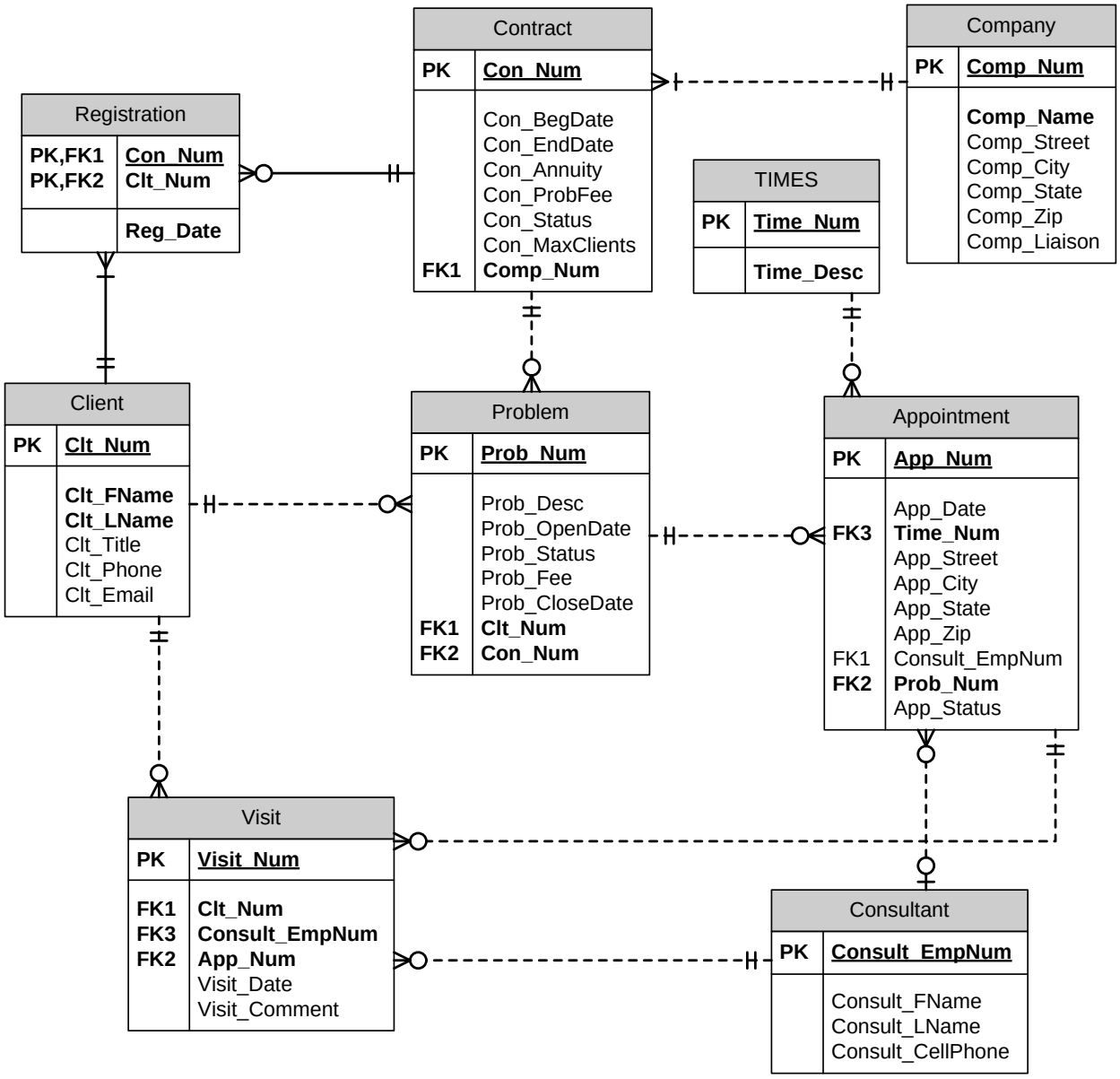

Figure 1. Implementation Data Model

COMPANY table

Attribute	Data Type	Keys	Constraints
Comp_Num	Integer Number	PK	Primary key
Comp_Name	Variable-length text up to 70 bytes		Cannot be Null
Comp_Street	Variable-length text up to 70 bytes		Cannot be Null
Comp_City	Variable-length text up to 50 bytes		Cannot be Null
Comp_State	Fixed-length text of 2 bytes		Cannot be Null
Comp_Zip	Fixed-length text of 5 bytes		Cannot be Null
Comp_Liaison	Variable-length text up to 50 bytes		

CONTRACT table

Attribute	Data Type	Keys	Constraints
Con_Num	Integer Number	PK	Primary key
Con_BegDate	Date		Cannot be Null
Con_EndDate	Date		
Con_Annuity	Currency		Cannot be Null
Con_ProbFee	Currency		Cannot be Null
Con_Status	Variable-length text up to 15 bytes		Cannot be Null
Con_MaxClients	Integer number		
Comp_Num	Integer number	FK	Cannot be Null, References table COMPANY

CLIENT table

Attribute	Data Type	Keys	Constraints
Clt_Num	Integer number	PK	Primary key
Clt_FName	Variable-length text up to 20 bytes		Cannot be Null
Clt_LName	Variable-length text up to 20 bytes		Cannot be Null
Clt_Title	Variable-length text up to 70 bytes		
Clt_Phone	Variable-length text up to 12 bytes		Cannot be Null
Clt_Email	Variable-length text up to 70 bytes		

REGISTRATION table

Attribute	Data Type	Keys	Constraints
Con_Num	Integer number	PK, FK	Primary key, References table CONTRACT
Clt_Num	Integer number	PK, FK	Primary key, References table CLIENT
Reg_Date	Date		Default sysdate, Cannot be Null

PROBLEM table

Attribute	Data Type	Keys	Constraints
Prob_Num	Integer number	PK	Primary key
Prob_Desc	Variable-length text up to 150 bytes		Cannot be Null
Prob_OpenDate	Date		Default sysdate
Prob_Status	Variable-length text up to 20 bytes		Cannot be Null
Prob_Fee	Currency		Cannot be Null
Prob_CloseDate	Date		
Clt_Num	Integer number	FK	Cannot be Null, References table CLIENT
Con_Num	Integer number	FK	Cannot be Null, References table CONTRACT

CONSULTANT table

Attribute	Data Type	Keys	Constraints
Consult_EmpNum	Integer number	PK	Primary key
Consult_FName	Variable-length text up to 20 bytes		Cannot be Null
Consult_LName	Variable-length text up to 20 bytes		Cannot be Null
Consult_Cellphone	Variable-length text up to 12 bytes		

TIMES table

Attribute	Data Type	Keys	Constraints
Time_Num	Integer number	PK	Primary key
Time_desc	Variable-length text up to 30 bytes		Cannot be Null

APPOINTMENT table

Attribute	Data Type	Keys	Constraints
App_Num	Integer number	PK	Primary key
App_Date	Date		Cannot be Null
Time_Num	Integer number		Cannot be Null
App_Street	Variable-length text up to 70 bytes		Cannot be Null
App_City	Variable-length text up to 70 bytes		Cannot be Null
App_State	Fixed-length text of 2 bytes		Cannot be Null
App_Zip	Fixed-length text of 5 bytes		
Consult_EmpNum	Integer number	FK	References table CONSULTANT
Prob_Num	Integer number	FK	Cannot be Null, References table PROBLEM
App_Status	Variable-length text up to 120 bytes		Default value is 'OPEN'

VISIT table

Attribute	Data Type	Keys	Constraints
Visit_Num	Integer number	PK	Primary key
Visit_Date	Date		Cannot be Null, Default sysdate
Visit_Comment	Variable-length text up to 200 bytes		
Clt_Num	Integer number	FK	Cannot be Null, References table CLIENT
Consult_EmpNum	Integer number		Cannot be Null, References table CONSULTANT
App_Num	Integer number		Cannot be Null, References table APPOINTMENT

Figure 2. Create Tables

COMPANY table

Comp_Num	Comp_Name	Comp_Street	Comp_City	Comp_ State	Comp_ Zip	Comp_ Liaison
1	Mardi Gras Supply	103 Beal Street	Memphis	TN	36450	Joey Barton
2	Guitars Unlimited	415 W. 2nd Avenue	Nashville	TN	37145	
3	Jones Discount Helmets	4110 E. 42nd Street, Suite 10	Nashville	TN	37145	
4	Grand Ball Dance Studios	1523 3rd Avenue	Nashville	TN	34560	Lorna Johns
5	Leftman and Schwartz	1020 Elm Plaza	Hendersonville	TN	34410	Rebecca Leftman
6	Western Wear LLC	114 Park Street	Nashville	TN	34560	Norman Adams
7	The Tradition	332 Barclay Lane	Florence	AL	35650	Steven Bartlett
8	Pet Menagerie	101 Bear Cove Drive	Nashville	TN	34540	

CONTRACT table

Con_ Num	Con_ Begdate	Con_ Enddate	Con_ Annuity	Con_ Probfee	Con_ Status	Con_ Maxclients	Comp_ Num
1	22-JUN-2006	21-JUN-2008	10000	20	OPEN	35	1
2	01-FEB-2004	01-FEB-2006	5000	50	CLOSED	15	2
3	16-MAY-2003	16-MAY-2006	40000	0	CLOSED	60	5
4	22-JUN-2006	21-JUN-2008	5000	100	OPEN	20	3
5	22-JUN-2006	21-JUN-2008	20000	30	OPEN	5	4
6	22-JUN-2006	21-JUN-2008	30000	40	OPEN	100	8
7	22-JUN-2006	21-JUN-2008	20000	20	OPEN	100	5
8	01-AUG-2006		100000	0	PENDING	350	6
9	22-JUN-2006	21-JUN-2008	10000	20	OPEN	3	2
10	22-JUN-2006	21-JUN-2008	2000	120	OPEN	30	7

CLIENT table

Clt_Num	Clt_Fname	Clt_Lname	Clt_Title	Clt_Phone	Clt_Email
1	Sarah	Chiang	Project Manager	615-555-3456	schiang@nomail.com
2	Anita	Lopez	Sales Agent	615-555-5908	arlopez@nomail.com
3	Gavin	Duncan	Purchasing Officer	615-555-7500	duncan.gavin@nomail.com
4	Peter	Walters	Office Manager	615-555-1300	petew@nomail.com
5	Samantha	Roberts		256-555-6123	
6	Jessica	Stevens	Regional Sales Manager	615-555-0986	stevensj@nomail.com
7	Robert	Valdez	Manager	615-555-5446	valdez@nomail.com
8	Bill	Miles		256-555-9876	billmiles@nomail.com
9	Glenn	Philips	Compliance Manager	615-555-7656	
10	Chase	Liu		615-555-1550	liu.chase@nomail.com
11	John	Gray	Product Engineer	615-555-5432	jgray@nomail.com
12	Paul	Lowe	Sales	615-555-3456	
13	Richard	Hunt		615-555-8901	rhunt@nomail.com
14	Beverly	Harris	Office Assistant	615-555-6000	blh2@nomail.com
15	Jodi	McManus	Assistant Director Human Resources	615-555-8003	hr@nomail.com

REGISTRATION table

Con_Num	Clt_Num	Reg_Date
1	5	23-JUN-2006
2	1	02-FEB-2004
2	2	03-MAR-2005
3	3	20-AUG-2004
3	10	19-MAY-2003
4	4	23-JUN-2006
5	7	23-JUN-2006
6	6	23-JUN-2006
7	15	25-JUN-2006
7	3	24-JUN-2006
7	10	24-JUN-2006
6	9	23-JUN-2006
9	2	23-JUN-2006
9	11	23-JUN-2006
10	5	23-JUN-2006
10	8	23-JUN-2006
9	1	22-JUN-2006
1	12	23-JUN-2006
4	13	23-JUN-2006
5	14	23-JUN-2006

PROBLEM table

Prob_ Num	Prob_Desc	Prob_ Opendate	Prob_ Status	Prob_ Fee	Prob_ Closedate	Clt_ Num	Con_ Num
1	PC - Hard drive dead	24-JUN-2006	OPEN	20		1	9
2	Xerox Copier - Document Feeder gear stripped	24-JUN-2006	OPEN	20		3	7
3	PC - Monitor dead	24-JUN-2006	OPEN	100		4	4
4	Mac - cannot obtain IP address	24-JUN-2006	OPEN	20		5	1
5	Fax - drum unit failure	24-JUN-2006	OPEN	20		1	9
6	Projector - cooling fan not working	24-JUN-2006	OPEN	30		7	5
7	PC - Laptop - boot failure	24-JUN-2006	CLOSED	20	26-JUN-2006	10	7
8	Xerox Copier - drum unit failure	24-JUN-2006	OPEN	120		12	10

APPOINTMENT table

App_ Num	App_ Date	Time_ Num	App_ Street	App_ City	App_ State	App_ Zip	Consult_ Empnum	Prob_ Num	App_ Status
1	24-JUN-2006	1	415 W. 2nd Avenue	Nashville	TN	37145	1	1	UNRESOLVED
2	24-JUN-2006	2	1020 Elm Plaza	Hendersonville	TN	34410	3	2	UNRESOLVED
3	25-JUN-2006	1	4110 E. 42nd Street, Suite 10	Nashville	TN	37145	1	3	UNRESOLVED
4	25-JUN-2006	2	103 Beal Street	Memphis	TN	36450	1	4	OPEN
5	26-JUN-2006	3	415 W. 2nd Avenue	Nashville	TN	37145	2	5	OPEN
6	26-JUN-2006	3	1523 3rd Avenue	Nashville	TN	34560	3	6	OPEN
7	26-JUN-2006	1	1020 Elm Plaza	Hendersonville	TN	34410	4	7	UNRESOLVED
8	26-JUN-2006	4	332 Barclay Lane	Florence	AL	35650	3	8	OPEN
9	26-JUN-2006	1	4110 E. 42nd Street, Suite 10	Nashville	TN	37145	2	3	OPEN
10	27-JUN-2006	4	103 Beal Street	Memphis	TN	36450	1	4	OPEN
11	27-JUN-2006	2	1020 Elm Plaza	Hendersonville	TN	34410	4	7	CANCELLED BY CLIENT
12	27-JUN-2006	3	415 W. 2nd Avenue	Nashville	TN	37145	3	1	OPEN
13	25-JUN-2006	1	1020 Elm Plaza	Hendersonville	TN	34410	4	2	OPEN

CONSULTANT table

Consult_Empnum	Consult_Fname	Consult_Lname	Consult_Cellphone
1	Perry	Waters	615-317-1234
2	Jackson	Summers	615-317-1235
3	Kari	Patel	615-317-1236
4	Lynda	Humboldt	615-317-1237
5	Stacie	Riviera	615-317-1238

TIMES table

Time_Num	Time_Desc
1	BEFORE 10AM
2	10AM – BEFORE NOON
3	NOON - 2PM
4	AFTER 2PM

VISIT table

Visit_ Num	Visit_ Date	Visit_Comment	Clt_ Num	Consult_ Empnum	App_ Num
1	24-JUN-2006	Arrived 9am. Client unavailable.	1	1	1
2	24-JUN-2006	Arrived 10:30am. Need new feeder plate in addition to the gears.	3	3	2
3	25-JUN-2006	Arrived 11am. Incorrect monitor specified -- return visit scheduled	4	1	3
4	25-JUN-2006	Arrived 8am. Recreated MBR.	10	4	7

Figure 3. Insert Data

2.2.6 Create Sequences.

Create two sequences: one named APP_NUM_SEQ, and the other named CLIENT_NUM_SEQ. Both sequences should start at 100, increment by 1, and not cache numbers.

2.2.7 Create Stored Procedures.

Write the code to create the following stored procedures. These procedures will be used by application programming teams; therefore, they must be named exactly as specified. Any changes to the data that are made to test the stored procedures must be rolled back.

1. Create a stored procedure (NEW_APP) for entering records into the Appointment table. The procedure must meet these requirements.
   a.) The user will specify: date, time number, problem number, and consultant employee number.
   b.) Check that the consultant is not already scheduled for an open appointment on that date during that time. If the consultant is already scheduled, then output a message to the user indicating the conflict (for example, "Schedule Conflict for Consultant John Smith") and do not insert the record.
   c.) If there is not a conflict, then use the APP_NUM_SEQ sequence to generate a new appointment number and insert the record. Output a message to the user that the appointment was entered (for example, "Appointment scheduled successfully").
2. Create a stored procedure (CLOSE_PROBLEM) for updating records in the Problem table. The procedure must meet these requirements.
   a.) The user will specify the problem number.
   b.) Check to see if the problem already has a status of "CLOSED." If the status is already closed, then output a message to the user indicating that the problem is already closed.
   c.) If the problem is not already closed, the set the problem status to "CLOSED," set the problem close date to the current date, and set the appointment status of any open appointments in the Appointment table that are associated with that problem to "CANCELLED BY CLIENT."
3. Create a stored procedure (NEW_CLIENT) for inserting records into the Client and Registration tables. The procedure must meet these requirements.
   a.) The user will supply the client first name, client last name, title, phone number, email address, and contract number.
   b.) Check to see if the number of users already registered under that contract number is greater than or equal to the contract's maximum number of clients. If the maximum number of clients for that contract has already been reached, then output a message to the user indicating that the "Maximum number of Clients already registered."
   c). If the maximum number of clients has not been reached, then use the CLIENT_ NUM_SEQ sequence (which you previously created) to generate a new client number.
   d.) Insert the new client information into the Client table.

e.) Using the same client number generated, insert the new registration into the Registration table using the current system date as the registration date.
f.) Output a message to the user that the client was successfully registered.

2.2.8 Create Views.

Database views can be used to restrict a user's ability to see some attributes within a table, or to shield users from the complexity of the query necessary to produce a given output. Views can be materialized or non-materialized. A materialized view creates a copy of the data returned by the underlying SELECT query, while a non-materialized view does not. Write the SQL code to create the following database views. Since the data needed for the views are available in the tables within the same database that the views will exist in, non-materialized views should be created.

1. Create a view called BILL that will display the contract number, contract begin date, and the total amount to be billed on that contract (labeled "Bill Total", formatted in dollar format) for all contracts that have been used to report a problem. The total bill is calculated as the annuity amount plus the sum of the actual problem fees that have been charged on all problems reported on that contract.
2. Create a view called OPENCONTRACT that will display the contract number and the number of clients that can still register on each contract (labeled "Client Slots Available"). The number of client slots is calculated as the maximum number of clients that can register on the contract minus the number of clients that have already registered on that contract.
3. Create a view called CONTRACTUSE that will display contract number, contract annuity amount, contract problem fee, problem description, problem fee charged, and problem status for all contracts, including contracts that have not had any problems reported. The result should be sorted in descending order by contract number.

AUTHOR BIOGRAPHY

Steven A. Morris is an Associate Professor of Computer Information Systems at Middle Tennessee State University. He received his Ph.D. in Management Information Systems from Auburn University. Dr. Morris conducts research in the cognitive factors that influence information systems design and adoption, organizational design, and various system analysis and design issues. His primary teaching focus is on database design, implementation, and management at both the undergraduate and graduate levels.

Information Systems & Computing
Academic Professionals

STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN 1055-3096