

---

# ICATS: Final Report

---

**L546: User Centered Database Design**

Anna Kaziunas  
Mark Millard

April 30, 2004

# Table of Contents

## Section 1: ICATS: Interface Usability Study

- 1.0 Purpose of the Study ..... 4
- 2.1 System Purpose ..... 4
- 2.2 System Overview..... 5
- 2.3 Usability Objectives ..... 5
- 2.4 Intended Audience ..... 5
- 3.0 Method..... 5
- 3.1 Subject Profiles ..... 5
- 3.2 Tasks ..... 5
- Table A – User Tasks ..... 5
- Table B – Post-Test Questionnaire Results ..... 7
- Table C – Summary of Results and Recommendations..... 7
- 4.0 Findings And Recommendations..... 8
- 4.1 Observation 1: Office Location Menu ..... 8
- 4.2 Observation 2: Logout Text..... 8
- 4.3 Observation 3: Repeating Events ..... 9
- 4.4 Observation 4: Calendar Feature ..... 9
- 4.5 Observation 5: Label Confusion ..... 9
- 4.6 Observation 6:University Email Extension ..... 9
- 4.7 Observation 7: Report Display Functions ..... 9
- 5.0 Conclusion..... 9
- Appendix A – Testing Procedure Overview ..... 11
- Appendix B – Post Test Questionnaire..... 12
- Appendix C – Debriefing Questions ..... 13

## Section 2: ICATS: Database Implementation & Test Report

- 1.0 Create Files (tables, constraints, sequences) ..... 17
- 2.0 Table Population (insert script) File ..... 23
- 3.0 Creation Spool (table, constraints, sequences) File ..... 31
- 3.1 Create Table Spool..... 31
- 3.2 Create Constraints Spool ..... 32
- 3.3 Create Sequences Spool ..... 36
- 4.0 Table Populate (insert) Spool File ..... 37
- 5.0 Table Constraint Testing SQL File..... 43
- 6.0 Table Constraint Testing SQL File..... 45
- 7.0 Query Generation File (all SQL commands) ..... 47
- 8.0 Query Results Spool File..... 51

## SECTION 1

---

# ICATS: Interface Usability Study

---

Staff Interface: <http://ella.slis.indiana.edu/~akaziuna/ICO/>

## 1.0 Usability Testing Executive Summary

The following study analyzed the usability of the ICATS database web interface prototype.

The ICATS usability testing was conducted with three subjects, all of whom are members of the ICO staff. The purpose of this study was to identify usability problems with the interface and to determine users' levels of satisfaction with the site. Participants performed six different tasks during a 30 minute session (each) during the week of April 29 in the ICO office at the School of Education. The tests resulted in these findings:

Positive Aspects about ICATS interface:

1. The interface look and feel were well received.
2. The participants were able to perform tasks very quickly and easily.
3. The forms had the proper options for recording actual client sessions / activities and worked well in practice.

Areas in the ICATS interface that need improvement:

1. The login option needs to be made to look more 'clickable' and different from the page headings.
2. A calendar feature was lacking in the prototype.
3. A repeating events feature was not present in the prototype.
4. Report printing and record linking features were absent in the prototype.

Based on the usability problems identified in this test, the usability team recommends:

- Making the login option into a button to show that it is 'clickable'.
- Adding a calendar function.
- Adding a repeating events function.
- Adding print and record linking features to the 'create reports' section of the interface.

## 2.0 Purpose of the Study

### 2.1 System Purpose

The Instructional Consulting Office (ICO) located in the School of Education (SOE) provides instructional and technology consulting to Faculty and Associate Instructors throughout the SOE. Currently, no system exists for the purposes of tracking consultant activities, nor does any real process exist in any form. At most the current process for tracking consultant activities is left up to the individual consultants. When talking with individual consultants, they each admit that they have no real process for keeping track of their activities. Annual reporting then suffers as each consultant attempts to list out all of his or her activities for the previous term.

### 2.2 System Overview

The ICATS database system will provide ICO consultants with a process, and a system for entering, querying, and reporting consulting and workshop activities. The system will be web-based, and accessible on the World Wide Web, via any available web browser. ICO employees will have the ability to add, edit, and in some cases delete project activity, client, and workshop data. The system will also increase the speed and accuracy of periodic reporting.

### 2.3 Usability Objectives

The testing was designed to evaluate the following:

- Can ICO consultants quickly and easily navigate the interface to accomplish their daily tasks?
- Is the interface navigation and labeling easy to understand?
- Does the staff like the look and feel of the interface?

### 2.4 Intended Audience

This system was designed specifically for the needs of the ICO. It is intended for use by the consultants, coordinator and director of the ICO only. At any given time there are two GA consultants, one coordinator and one director in the ICO office.

## 3.0 Method

### 3.1 Subject Profiles

The participants were two ICO consultants and the ICO coordinator, three participants total. They were contacted through a member of the ICATS project team who is also an ICO consultant. All of the participants were comfortable using the internet, as they work in a computer related field. All of the participants were male. There was a combination of a full time employee and two student GA employees.

### 3.2 Tasks

Each of the three users was given six tasks to complete. The login and logout tasks were given first and last, respectively. All other tasks were given in a random order on both a Mac and a PC at the ICO office at the School of Education, which has access to the campus network. The browsers the users chose to use were Safari (Mac) and Internet Explorer (PC). The tasks are as follows:

**Table A – User Tasks**

1.	Log in to the interface. Your username is guest. Your password is 3guestio.
2.	A client (Tom Jefferson) comes in to the office and informs you that they have changed their phone number to (812) 333-3455. How would you update their information?
3.	A client (Susan Anthony) came in to the office by appointment to meet with you to discuss an upcoming project involving scanning that she would like

	help putting together. She needs help with scanning documents on a bi-monthly basis in order to get some course materials online. After the client leaves, how do you record this interaction?
4.	A new consultant has just been hired (Joe Everyman). He will have administrator privileges on the system. His email is <a href="mailto:jeverey@indiana.edu">jeverey@indiana.edu</a> and his phone is (812)-333-4452. His username will be set as jevery and his initial password will be woffe4i. How would you add Joe's information to the database?
5.	You need view all sessions that occurred regarding VHS dubbing for a report you need to create. Where will you go to find this information?
6.	Log out of the interface.

Test subjects were asked to think aloud while they performed their tasks. The number of clicks and scrolls were noted, however the users were typing the question information into the forms in order so simulate the actual client experience. Because the test took place in the ICO office and not a laboratory environment, keystrokes could not be captured.)

Comments that the subjects made while thinking aloud were recorded. The user also filled out a debriefing questionnaire after the test. (See Appendix A) The test subjects were then interviewed and asked direct questions about the interface.

More details of the testing procedure are found in tables A and B, as well as Appendix A and Appendix B.

Known problems, recommendations for solutions and their significance are discussed in the Results and Recommendations section of this report.

## Results

Overall, the results were very favorable. The participants were able to easily able to complete the tasks and the survey information (Table B) showed that overall, the participants felt that the interface was pleasing, the reports were simple to create, and overall navigation intuitive.

**Table B – Post-Test Questionnaire Results:**

Questions	Answer Options										
		1	2	3	4	5	6	7	N/A	Ave.	
Recording client interactions was	difficult						◆	◆ ◆		easy	6.7
Logging out of the system was	difficult							◆ ◆ ◆		easy	7.0
The terms used for navigation were	ambiguous							◆ ◆ ◆		precise	7.0
The way information was organized throughout the site was	confusing						◆	◆ ◆		clear	6.7
Your expectations for where the information was located on the site were	confusing						◆ ◆	◆		clear	6.3
The text size on each screen was	difficult to read						◆	◆ ◆		easy to read	6.7
The use of graphics and color, and on the interface was	pleasing							◆ ◆ ◆		not pleasing	7.0
The need for help was	frequent							◆ ◆ ◆		infrequent	7.0
The reports were easy to create	difficult						◆	◆ ◆		easy	6.7
Overall experience in using the site was	frustrating							◆ ◆	◆	enjoyable	7.0

## 4.0 Findings and Recommendations

Significance was determined by looking at the impact and frequency of occurrences. A summary of our recommendations is contained in Table C.

**Table C – Summary of Results and Recommendations**

Observation	Recommendation	Significance
The office location pull down menu needs to reflect real locations.	Add authentic locations to pull down menu.	Medium
The logout text does not look clickable because it is the same format/color and position as the page title.	Create a button to differentiate the logout option or change the look /color	Medium
All users suggested a repeating event feature.	Add a repeating event option to the record client session page.	High
All users suggested a calendar feature to go with the repeating event feature.	Perhaps this can be added in a later version of the interface.	Medium
Two users were a little confused about what the difference was between 'admin' and 'administration' in the user types section.	Re-evaluate the categories and consolidate them or explain what is meant by putting a description next to the menu.	High
One user repeatedly mentioned that it would be useful when adding email to clients and employee profiles to have the @Indiana.edu section filled in because all employees / clients are associated with the university.	Have the @Indiana.edu extension come up in the textbox as a default to save typing time.	Medium
One user wanted report display functions that could print and be cut and pasted.	Add extra functionality to the report section to meet consultants' needs.	High

### 4.1 Observation 1: Office Location Menu

The office location menu was not populated with the correct consulting locations in the mockup on which we ran the test (an oversight on the part of the interface designer). All participants pointed this out this oversight and it will be corrected in the actual interface.

### 4.2 Observation 2: Logout Text

There is too much similarity between the page headings (which are bold, red and in the left hand corner of the page under the navigation and above the form. The logout option is also red, bold and in a similar position to the page heading, but it is on the right hand corner below the navigation. The test participants mentioned that they thought it would be

beneficial to differentiate the page title from the logout option by making the logout a different color or a button to make it appear clickable.

### **4.3 Observation 3: Repeating Events**

Each user independently suggested that a repeating events feature would be very beneficial for the system. This feature would enable consultants to more realistically add repeating client appointments and ongoing projects. This feature is currently being explored for possible final system implementation

### **4.4 Observation 4: Calendar Feature**

The participants also suggested that in combination with the repeating events feature, a calendar feature would be very helpful in entering data and tracking appointments. This option is currently being explored for possible final system implementation.

### **4.5 Observation 5: Label Confusion**

There was a little confusion for one user on the 'update employee information' page between the employee roles menu and option of 'Administrator' and the user type of 'Admin'. This will be made clearer in the final system

### **4.6 Observation 6: University Email Extension**

It was suggested the it would save time if the university email extension of '@indiana.edu' would automatically be filled in on the text areas where an email address was required. This would be useful because all clients and staff have IU email addresses.

### **4.7 Observation 7: Report Display Functions**

At the time of testing, the report display functions were not yet functional and there was a dummy results page in place. However, users suggested that some helpful report display functions would be:

- A clickable records listing to each individual record in the report.
- A printable view.
- A view where text can be cut and pasted easily.

## **5.0 Conclusion**

The goal of the testing, as stated at the beginning of this report, was to examine how effectively and efficiently an ICO staff member is able to record, update and add information to the ICATS database using the graphical interface.

The testing was conducted with the following objectives in mind:

- Can ICO consultants quickly and easily navigate the interface to accomplish their daily tasks?
- Is the interface navigation and labeling easy to understand?
- Does the staff like the look and feel of the interface?

The purpose of ICATS system is to provide the ICO staff with a quick and easy way to track client information for reporting purposes. Although some reporting and calendar functions were absent in the prototype, users were easily able to complete the recording and updating tasks.

Overall, the test results showed that overall the interface meet the needs of the ICO staff, however, additional features such as more report functionality, calendar functions and repeating event functions are highly desirable. The usability team recommends that these functions should be added to the interface in order to maximize both system and ICO staff performance.

## **Appendix A - Testing Procedure Overview**

### **Preliminary Stage**

- The moderator explains the testing protocol to the participant.

### **Task Performance Stage**

- The moderator reads the script with instructions on how the tasks should be performed explaining the user's rights and the usability team's obligations to privacy.
- The moderator gives the packet with the tasks to the user.
- The user proceeds through each task while thinking aloud.
- Once all the tasks were completed, the moderator gives the user a post-questionnaire.

### **Post-testing Stage**

- The user completes the post-questionnaire.
- The moderator interviews the user and questions them about specific critical points noted time during the tasks.
- Moderator thanks the user for participating.

## Appendix B – Post-Test Questionnaire

(Please circle only one answer for each question)

Questions	Answer options		
Recording client interactions was	difficult	1 2 3 4 5 6 7 N/A	easy
Logging out of the system was	difficult	1 2 3 4 5 6 7 N/A	easy
The way information was organized throughout the site was	ambiguous	1 2 3 4 5 6 7 N/A	precise
Your expectations for where the information was located on the site were	confusing	1 2 3 4 5 6 7 N/A	clear
The text size on each screen was	difficult to read	1 2 3 4 5 6 7 N/A	easy to read
The use of graphics and color on the interface was	not pleasing	1 2 3 4 5 6 7 N/A	pleasing
The text size on each screen was	difficult to read	1 2 3 4 5 6 7 N/A	easy to read
The need for help was	frequent	1 2 3 4 5 6 7 N/A	infrequent
The reports were easy to create	difficult	1 2 3 4 5 6 7 N/A	easy
Overall experience in using the site was	frustrating	1 2 3 4 5 6 7 N/A	easy

## Appendix C – Debriefing Questions

### **TASK 1:**

Log in to the interface.

#### **Task 1 Questions:**

Was the login screen what you expected?

---

### **TASK 2:**

A client (Tom Jefferson) comes in to the office and informs you that they have changed their phone number to (812) 333-3455. How would you update their information?

#### **Task 1 Question:**

*Did you experience any difficulties in updating the phone number? Why or why not? Why did you choose the approach you took? What made you think that it would work?*

---

### **TASK 3:**

A client (Susan Anthony) came in to the office by appointment to meet with you to discuss an upcoming project involving scanning that she would like help putting together. She needs help with scanning documents on a bi-monthly basis in order to get some course materials online.

After the client leaves, how do you record this interaction?

#### **Task 2 Questions:**

*Were you able to record the interaction easily? Why or why not?*

---

### **TASK 3:**

A new consultant has just been hired (Joe Everyman). He will have administrator privileges on the system. His email is [jeverey@indiana.edu](mailto:jeverey@indiana.edu) and his phone is (812)-333-4452. His username will be set as jevery and his initial password will be woffe4i.

How would you add Joe's information to the database?

**TASK 3 QUESTIONS**

*Was it difficult to add the new employee? Why or why not?*

---

**TASK 5:**

You need view all sessions that occurred regarding VHS dubbing for a report you need to create. Where will you go to find this information?

**Task 4 Questions**

*Was the report easy to create? Why or why not? Did you have any trouble understanding the label? Why do you think it was difficult / easy?*

---

**TASK 6:**

Log out of the interface.

**Task 5 Questions:**

Where did you look for the logout? Why did you think it would be there?

---

**GENERAL IMPRESSIONS:**

*What were your general impressions of the ICO Staff interface?*

---

**NAVIGATION:**

*What did you think about the site's navigation in general? Was it easy to find the information you were looking for? Were the items you were searching for located under the categories you expected? Was it easier or more difficult than you expected? Was it relatively easy to use this site or relatively difficult?*

---

**APPEARANCE / LAYOUT**

Was the site visually consistent and clear? Do you think categories and concepts were explicitly stated and consistently maintained? Was the labeling clear? Were the colors complementary or did you find them distracting?

---

**PARTICIPANT'S RECOMMENDATIONS**

*Do you have any recommendations or suggestions for the ICO employee interface?*

## **SECTION 2**

---

# **ICATS: Database Implementation & Test Report**

---

## Section 2-- Database Implementation & Test Report:

The database implementation of the ICATS project has been a success. A full implementation of the database exists and is ready for future expansion of business needs. In terms of project challenges, the work on creating constraints (both primary key and foreign constraints) was probably our biggest challenge. This wasn't as much a conceptual or technical challenge, but more about learning the process. We were essentially having a chicken & egg problem with constraints trying to be applied to tables that hadn't been created yet, as well as a few data type incompatibility errors, which were easily solved by double checking all our data types and making sure they all matched up.

In regard to successfully getting our constraints created, we decided to separate our table create script, and our create constraint script. This allowed our create table script to complete successfully. Then we were able to go back systematically go back and apply constraints in the proper order, in terms of their parent-child relationships. As a result, all tables and constraints have been created successfully.

Having had some issues with inserting our data at first. Once we again remembered that we needed to insert with respect to the parent-child relationships and ordered our insert script accordingly, inserting all of our data was successful in one pass.

All queries and constraints have been fully tested and are discussed in the appropriate sections below. In addition, sequences have been successfully created and are being utilized with our numerical primary keys to maintain data integrity.

### 1.0- Create Files (tables, constraints, sequences): /

Purpose & Method	We decided to separate our table create script, and our create constraint script. This allowed our create table script to complete successfully, and then we were able to go back and apply constraints in the proper order in tune with parent-child relationships. All tables were created successfully.
------------------	--

```
--
-- ICATS Database Model Create Scripts
-- Mark Millard and Anna Kaziunas
--
```

```
--
-- TABLE: SERVICE
--
CREATE TABLE SERVICE(
```

```

        SERV_ID                NUMBER(6) NOT NULL,
        LOCATION_ID            NUMBER(6) NOT NULL,
        SERV_TYPE_ID           VARCHAR2(6) NOT NULL,
        SERV_CATG_ID           NUMBER(6) NOT NULL,
        SERV_NAME               VARCHAR2(50),
        SERV_REQUEST_DATE      VARCHAR2(20),
        SERV_COMPLETE_DATE     VARCHAR2(20),
        SERV_LENGTH             VARCHAR2(20),
        SERV_COMMENTS          VARCHAR2(2000)
    );

--
-- TABLE:  SERV_CATEGORY
--
CREATE TABLE SERV_CATEGORY(
    SERV_CATG_ID                NUMBER(6) NOT NULL,
    SERV_CATG_NAME              VARCHAR2(50)
);

--
-- TABLE:  SERV_TYPE
--
CREATE TABLE SERV_TYPE(
    SERV_TYPE_ID                VARCHAR2(6) NOT NULL,
    SERV_TYPE_NAME              VARCHAR2(70)
);

--
-- TABLE:  LOCATION
--
CREATE TABLE LOCATION(
    LOCATION_ID                NUMBER(6) NOT NULL,
    LOCATION_NAME              VARCHAR2(50),
    LOCATION_SIZE              VARCHAR2(20),
    LOCATION_ADDRESS           VARCHAR2(70),
    LOCATION_PHONE             VARCHAR2(20),
    LOCATION_ROOM              VARCHAR2(20),
    LOCATION_BUILDING          VARCHAR2(20)
);

--
-- TABLE:  SERV_REQUEST
--
CREATE TABLE SERV_REQUEST(
    CLIENT_ID                  NUMBER(6) NOT NULL,
    SERV_ID                    NUMBER(6) NOT NULL
```

```
);  
  
--  
-- TABLE: CLIENT  
--  
CREATE TABLE CLIENT(  
    CLIENT_ID NUMBER(6) NOT NULL,  
    DEPT_ID    VARCHAR2(6) NOT NULL,  
    CLIENT_FNAME VARCHAR2(30),  
    CLIENT_LNAME VARCHAR2(30),  
    CLIENT_USERNAME VARCHAR2(15),  
    CLIENT_PASSWD VARCHAR2(10),  
    CLIENT_EMAIL VARCHAR2(30),  
    CLIENT_PHONE VARCHAR2(10)  
);  
  
--  
-- TABLE: DEPARTMENT  
--  
CREATE TABLE DEPARTMENT(  
    DEPT_ID    VARCHAR2(6) NOT NULL,  
    DEPT_NAME  VARCHAR2(30)  
);  
  
--  
-- TABLE: SERV_ASSIGN  
--  
CREATE TABLE SERV_ASSIGN(  
    EMP_ID    NUMBER(6) NOT NULL,  
    SERV_ID   NUMBER(6) NOT NULL  
);  
  
--  
-- TABLE: EMPLOYEE  
--  
CREATE TABLE EMPLOYEE(  
    EMP_ID    NUMBER(6) NOT NULL,  
    EMP_FNAME VARCHAR2(30),  
    EMP_LNAME VARCHAR2(50),  
    EMP_USERNAME VARCHAR2(15),  
    EMP_PASSWD VARCHAR2(10),  
    EMP_ROLE_ID VARCHAR2(20) NOT NULL,  
    EMP_EMAIL VARCHAR2(20),  
    EMP_PHONE VARCHAR2(20)  
);
```

```
--
-- TABLE: EMP_TYPE
--
CREATE TABLE EMP_TYPE(
    EMP_ROLE_ID    VARCHAR2(20) NOT NULL,
    EMP_ROLE_NAME VARCHAR2(40)
);

--
-- ICATS-- Create (alter) Constraints Script
-- Mark Millard and Anna Kaziunas
--
--
-- TABLE:  LOCATION
--
ALTER TABLE LOCATION ADD(
    CONSTRAINT    location_locationid_pk    PRIMARY KEY    (LOCATION_ID)
);

--
-- TABLE:  SERV_TYPE
--
ALTER TABLE SERV_TYPE ADD(
    CONSTRAINT    servtype_servtypeid_pk    PRIMARY KEY    (SERV_TYPE_ID)
);

--
-- TABLE:  SERV_CATEGORY
--
ALTER TABLE SERV_CATEGORY ADD(
    CONSTRAINT    servcatg_servcatgid_pk    PRIMARY KEY    (SERV_CATG_ID)
);

--
-- TABLE: DEPARTMENT
--
ALTER TABLE DEPARTMENT ADD(
    CONSTRAINT    department_deptid_pk    PRIMARY KEY    (DEPT_ID)
);

--
```

```
-- TABLE: CLIENT
```

```
--  
ALTER TABLE CLIENT ADD(  
    CONSTRAINT    client_clientid_pk PRIMARY KEY (CLIENT_ID),  
    CONSTRAINT    client_deptid_fk   FOREIGN KEY   (DEPT_ID)  
REFERENCES DEPARTMENT (DEPT_ID)  
);
```

```
--  
-- TABLE: EMP_TYPE
```

```
--  
ALTER TABLE EMP_TYPE ADD(  
    CONSTRAINT    emp_type_emp_role_id_pk      PRIMARY KEY  
    (EMP_ROLE_ID)  
);
```

```
--  
-- TABLE: EMPLOYEE
```

```
--  
ALTER TABLE EMPLOYEE ADD(  
    CONSTRAINT    employee_emp_id_pk      PRIMARY KEY      (EMP_ID),  
    CONSTRAINT    employee_role_id_fk     FOREIGN KEY      (EMP_ROLE_ID)  
REFERENCES EMP_TYPE (EMP_ROLE_ID)  
);
```

```
--  
-- TABLE: SERVICE
```

```
--  
ALTER TABLE SERVICE ADD(  
    CONSTRAINT    service_servid_pk PRIMARY KEY (SERV_ID),  
    CONSTRAINT    service_locationid_fk FOREIGN KEY (LOCATION_ID)  
REFERENCES LOCATION (LOCATION_ID),  
    CONSTRAINT    service_servtypeid_fk FOREIGN KEY  
    (SERV_TYPE_ID) REFERENCES SERV_TYPE (SERV_TYPE_ID),  
    CONSTRAINT    service_servcatgid_fk FOREIGN KEY  
    (SERV_CATG_ID) REFERENCES SERV_CATEGORY (SERV_CATG_ID)  
);
```

```
--  
-- TABLE: SERV_ASSIGN
```

```
--  
ALTER TABLE SERV_ASSIGN ADD(  
    CONSTRAINT    serv_assign_pk PRIMARY KEY (EMP_ID,SERV_ID),  
    CONSTRAINT    servassign_empid_fk FOREIGN KEY (EMP_ID)  
REFERENCES EMPLOYEE (EMP_ID),
```

```
        CONSTRAINT servassign_servid_fk FOREIGN KEY (SERV_ID)
REFERENCES SERVICE (SERV_ID)
);
```

```
--
-- TABLE: SERV_REQUEST
--
ALTER TABLE SERV_REQUEST ADD(
    CONSTRAINT servrequest_pk PRIMARY KEY (CLIENT_ID,SERV_ID),
    CONSTRAINT servreq_clientid_fk FOREIGN KEY (CLIENT_ID)
REFERENCES CLIENT (CLIENT_ID),
    CONSTRAINT servreq_servid_fk FOREIGN KEY (SERV_ID)
REFERENCES SERVICE (SERV_ID)
);
```

```
--
-- ICATS-- Create Sequences Script
-- Mark Millard and Anna Kaziunas
--
```

```
--
-- SEQUENCE: LOCATION
--
CREATE SEQUENCE location_seq
    START WITH 1
    INCREMENT BY 1
    NOCACHE
    NOCYCLE;
```

```
--
-- SEQUENCE: SERV_CATEGORY
--
CREATE SEQUENCE servcatg_seq
    START WITH 1
    INCREMENT BY 1
    NOCACHE
    NOCYCLE;
```

```
--
-- SEQUENCE: CLIENT
--
CREATE SEQUENCE client_seq
    START WITH 1
    INCREMENT BY 1
    NOCACHE
```

```

NOCYCLE;

--
-- SEQUENCE: EMPLOYEE
--
CREATE SEQUENCE employee_seq
  START WITH 1
  INCREMENT BY 1
  NOCACHE
  NOCYCLE;

--
-- SEQUENCE: SERVICE
--
CREATE SEQUENCE service_seq
  START WITH 1
  INCREMENT BY 1
  NOCACHE
  NOCYCLE;

--
-- SEQUENCE: SERV_TYPE
--
CREATE SEQUENCE servtype_seq
  START WITH 1
  INCREMENT BY 1
  NOCACHE
  NOCYCLE;

```

## **2.0- Table Population (insert script) File-** /

Purpose	This section contains all of the insert statements that were used to add data
& Method	to our database. – comments indicate the start of each table.

```

--
-- ICATS-- Insert (populate) Script
-- Mark Millard and Anna Kaziunas
--
--
-- TABLE: LOCATION
--

```

```
INSERT INTO LOCATION ( location_id, location_name, location_size, location_address,
location_phone, location_room, location_building )
VALUES ( location_seq.NEXTVAL, 'SOE LTTS Lab', '23', '200 N. Rose Ave.', '856-1111',
'2010', 'WWWright Bldg' );
```

```
INSERT INTO LOCATION ( location_id, location_name, location_size, location_address,
location_phone, location_room, location_building )
VALUES ( location_seq.NEXTVAL, 'UITS Lab-1', '32', '200 N. Rose Ave.', '856-1121', '2012',
'WWWright Bldg' );
```

```
INSERT INTO LOCATION ( location_id, location_name, location_size, location_address,
location_phone, location_room, location_building )
VALUES ( location_seq.NEXTVAL, 'UITS Lab-2', '30', '200 N. Rose Ave.', '856-1122', '2015',
'WWWright Bldg' );
```

```
INSERT INTO LOCATION ( location_id, location_name, location_size, location_address,
location_phone, location_room, location_building )
VALUES ( location_seq.NEXTVAL, 'UITS Lab-3', '23', '200 N. Rose Ave.', '856-1123', '2021',
'WWWright Bldg' );
```

```
INSERT INTO LOCATION ( location_id, location_name, location_size, location_address,
location_phone, location_room, location_building )
VALUES ( location_seq.NEXTVAL, 'SOE Auditorium', '125', '200 N. Rose Ave.', '856-0001',
'1000', 'WWWright Bldg' );
```

```
INSERT INTO LOCATION ( location_id, location_name, location_size, location_address,
location_phone, location_room, location_building )
VALUES ( location_seq.NEXTVAL, 'IST Teaching Lab', '12', '200 N. Rose Ave.', '856-0002',
'2210', 'WWWright Bldg' );
```

```
INSERT INTO LOCATION ( location_id, location_name, location_size, location_address,
location_phone, location_room, location_building )
VALUES ( location_seq.NEXTVAL, 'CRLT Video Classroom', '21', '200 N. Rose Ave.', '856-
1001', '2112', 'WWWright Bldg' );
```

```
INSERT INTO LOCATION ( location_id, location_name, location_size, location_address,
location_phone, location_room, location_building )
VALUES ( location_seq.NEXTVAL, 'IST Conference Room', '25', '200 N. Rose Ave.', '856-
0003', '2220', 'WWWright Bldg' );
```

```
INSERT INTO LOCATION ( location_id, location_name, location_size, location_address,
location_phone, location_room, location_building )
VALUES ( location_seq.NEXTVAL, 'IST Classroom', '30', '200 N. Rose Ave.', '856-1100',
'2225', 'WWWright Bldg' );
```

--

```
-- TABLE: SERV_TYPE
--
INSERT INTO SERV_TYPE ( serv_type_id, serv_type_name )
VALUES ( servtype_seq.NEXTVAL, 'Project' );

INSERT INTO SERV_TYPE ( serv_type_id, serv_type_name )
VALUES ( servtype_seq.NEXTVAL, 'Workshop' );

--
-- TABLE: SERV_CATEGORY
--
INSERT INTO SERV_CATEGORY ( serv_catg_id, serv_catg_name )
VALUES ( servcatg_seq.NEXTVAL, 'Web Development' );

INSERT INTO SERV_CATEGORY ( serv_catg_id, serv_catg_name )
VALUES ( servcatg_seq.NEXTVAL, 'Digital Video' );

INSERT INTO SERV_CATEGORY ( serv_catg_id, serv_catg_name )
VALUES ( servcatg_seq.NEXTVAL, 'OnCourse' );

INSERT INTO SERV_CATEGORY ( serv_catg_id, serv_catg_name )
VALUES ( servcatg_seq.NEXTVAL, 'VHS Dubbing' );

INSERT INTO SERV_CATEGORY ( serv_catg_id, serv_catg_name )
VALUES ( servcatg_seq.NEXTVAL, 'Scanning' );

INSERT INTO SERV_CATEGORY ( serv_catg_id, serv_catg_name )
VALUES ( servcatg_seq.NEXTVAL, 'Graphic Design' );

INSERT INTO SERV_CATEGORY ( serv_catg_id, serv_catg_name )
VALUES ( servcatg_seq.NEXTVAL, 'Pedagogical Support' );

INSERT INTO SERV_CATEGORY ( serv_catg_id, serv_catg_name )
VALUES ( servcatg_seq.NEXTVAL, 'Evaluation' );

INSERT INTO SERV_CATEGORY ( serv_catg_id, serv_catg_name )
VALUES ( servcatg_seq.NEXTVAL, 'EvalOnline' );

INSERT INTO SERV_CATEGORY ( serv_catg_id, serv_catg_name )
VALUES ( servcatg_seq.NEXTVAL, 'SiteScape' );

INSERT INTO SERV_CATEGORY ( serv_catg_id, serv_catg_name )
VALUES ( servcatg_seq.NEXTVAL, 'Printing-Media' );

INSERT INTO SERV_CATEGORY ( serv_catg_id, serv_catg_name )
```

```
VALUES ( servcatg_seq.NEXTVAL, 'Other' );
```

```
--
```

```
-- TABLE: DEPARTMENT
```

```
--
```

```
INSERT INTO DEPARTMENT ( dept_id, dept_name )  
VALUES ( 'CEP', 'Counseling and Educational Psychology' );
```

```
INSERT INTO DEPARTMENT ( dept_id, dept_name )  
VALUES ( 'CI', 'Curriculum and Instruction' );
```

```
INSERT INTO DEPARTMENT ( dept_id, dept_name )  
VALUES ( 'DEAN', 'Deans Office' );
```

```
INSERT INTO DEPARTMENT ( dept_id, dept_name )  
VALUES ( 'HE', 'Higher Education' );
```

```
INSERT INTO DEPARTMENT ( dept_id, dept_name )  
VALUES ( 'IST', 'Instructional Systems Technology' );
```

```
INSERT INTO DEPARTMENT ( dept_id, dept_name )  
VALUES ( 'LE', 'Language Education' );
```

```
INSERT INTO DEPARTMENT ( dept_id, dept_name )  
VALUES ( 'LS', 'Learning Sciences' );
```

```
INSERT INTO DEPARTMENT ( dept_id, dept_name )  
VALUES ( 'OTHER', 'Other outside Departments' );
```

```
--
```

```
-- TABLE: CLIENT
```

```
--
```

```
INSERT INTO CLIENT ( client_id, dept_id, client_fname, client_lname, client_username,  
client_passwd, client_email, client_phone )  
VALUES ( client_seq.NEXTVAL, 'LE', 'Karen', 'Neuman', 'klneuman', '*****',  
'klneuman@indiana.edu', '856-0001' );
```

```
INSERT INTO CLIENT ( client_id, dept_id, client_fname, client_lname, client_username,  
client_passwd, client_email, client_phone )  
VALUES ( client_seq.NEXTVAL, 'CEP', 'Fred', 'Flinstone', 'fflin', '*****', 'fred@indiana.edu',  
'856-0001' );
```

```
INSERT INTO CLIENT ( client_id, dept_id, client_fname, client_lname, client_username,  
client_passwd, client_email, client_phone )
```

```
VALUES ( client_seq.NEXTVAL, 'HE', 'Barney', 'Rubble', 'brubble', '*****',
'barney@indiana.edu', '856-0001' );
```

```
INSERT INTO CLIENT ( client_id, dept_id, client_fname, client_lname, client_username,
client_passwd, client_email, client_phone )
VALUES ( client_seq.NEXTVAL, 'IST', 'Nancy', 'Smith', 'nsmith', '*****',
'nsmith@indiana.edu', '856-0001' );
```

```
INSERT INTO CLIENT ( client_id, dept_id, client_fname, client_lname, client_username,
client_passwd, client_email, client_phone )
VALUES ( client_seq.NEXTVAL, 'LE', 'George', 'Jones', 'gjones', '*****',
'jonesg@indiana.edu', '856-0001' );
```

```
INSERT INTO CLIENT ( client_id, dept_id, client_fname, client_lname, client_username,
client_passwd, client_email, client_phone )
VALUES ( client_seq.NEXTVAL, 'CEP', 'Hank', 'Hill', 'hnhill', '*****', 'hillh@indiana.edu',
'856-0001' );
```

```
INSERT INTO CLIENT ( client_id, dept_id, client_fname, client_lname, client_username,
client_passwd, client_email, client_phone )
VALUES ( client_seq.NEXTVAL, 'IST', 'Tim', 'Burners-Lee', 'tblee', '*****',
'timlee@indiana.edu', '856-0001' );
```

```
INSERT INTO CLIENT ( client_id, dept_id, client_fname, client_lname, client_username,
client_passwd, client_email, client_phone )
VALUES ( client_seq.NEXTVAL, 'LE', 'Sue', 'Hendler', 'suehend', '*****',
'hendler@indiana.edu', '856-0001' );
```

```
INSERT INTO CLIENT ( client_id, dept_id, client_fname, client_lname, client_username,
client_passwd, client_email, client_phone )
VALUES ( client_seq.NEXTVAL, 'CEP', 'Sarah', 'Burton', 'sburton', '*****',
'burtions@indiana.edu', '856-0001' );
```

```
INSERT INTO CLIENT ( client_id, dept_id, client_fname, client_lname, client_username,
client_passwd, client_email, client_phone )
VALUES ( client_seq.NEXTVAL, 'LE', 'Leona', 'Helms', 'helmsl', '*****',
'leona@indiana.edu', '856-0001' );
```

```
--
```

```
-- TABLE: EMP_TYPE
```

```
--
```

```
INSERT INTO EMP_TYPE ( emp_role_id, emp_role_name )
VALUES ( 'ADM', 'Administrator' );
```

```
INSERT INTO EMP_TYPE ( emp_role_id, emp_role_name )
```

```
VALUES ( 'CON', 'Consultant' );
```

```
INSERT INTO EMP_TYPE ( emp_role_id, emp_role_name )  
VALUES ( 'CRD', 'Coordinator' );
```

```
INSERT INTO EMP_TYPE ( emp_role_id, emp_role_name )  
VALUES ( 'DIR', 'Director' );
```

```
--
```

```
-- TABLE: EMPLOYEE
```

```
--
```

```
INSERT INTO EMPLOYEE ( emp_id, emp_fname, emp_lname, emp_username,  
emp_passwd, emp_role_id, emp_email, emp_phone )  
VALUES ( employee_seq.NEXTVAL, 'Mark', 'Millard', 'mmillard', '*****', 'CON',  
'@indiana.edu', '856-8063' );
```

```
INSERT INTO EMPLOYEE ( emp_id, emp_fname, emp_lname, emp_username,  
emp_passwd, emp_role_id, emp_email, emp_phone )  
VALUES ( employee_seq.NEXTVAL, 'Chris', 'Essex', 'cessex', '*****', 'CRD',  
'cessex@indiana.edu', '856-8409' );
```

```
INSERT INTO EMPLOYEE ( emp_id, emp_fname, emp_lname, emp_username,  
emp_passwd, emp_role_id, emp_email, emp_phone )  
VALUES ( employee_seq.NEXTVAL, 'Seak-zoon', 'Roh', 'roh', '*****', 'CON',  
'sroh@indiana.edu', '856-8064' );
```

```
INSERT INTO EMPLOYEE ( emp_id, emp_fname, emp_lname, emp_username,  
emp_passwd, emp_role_id, emp_email, emp_phone )  
VALUES ( employee_seq.NEXTVAL, 'Karen', 'Hallett', 'hallett', '*****', 'DIR',  
'hallett@indiana.edu', '856-8409' );
```

```
INSERT INTO EMPLOYEE ( emp_id, emp_fname, emp_lname, emp_username,  
emp_passwd, emp_role_id, emp_email, emp_phone )  
VALUES ( employee_seq.NEXTVAL, 'Admin', 'Administrator', 'admin', '*****', 'ADM',  
'icy@indiana.edu', '856-8409' );
```

```
--
```

```
-- TABLE: SERVICE
```

```
--
```

```
INSERT INTO SERVICE ( serv_id, location_id, serv_type_id, serv_catg_id, serv_name,  
serv_request_date, serv_complete_date, serv_length, serv_comments )  
VALUES ( service_seq.NEXTVAL, '1', '1', '1', 'website for P540', '08/10/2003', '08/10/2003',  
'walk-in', 'These are some comments 01' );
```

```
INSERT INTO SERVICE ( serv_id, location_id, serv_type_id, serv_catg_id, serv_name,  
serv_request_date, serv_complete_date, serv_length, serv_comments )
```

```
VALUES ( service_seq.NEXTVAL, '1', '1', '1', 'web help', '10/20/2003', '10/20/2003', 'walk-in',
'These are some comments 02' );
```

```
INSERT INTO SERVICE ( serv_id, location_id, serv_type_id, serv_catg_id, serv_name,
serv_request_date, serv_complete_date, serv_length, serv_comments )
VALUES ( service_seq.NEXTVAL, '1', '1', '3', 'helped fred with Oncourse', '11/12/2003',
'11/12/2003', 'walk-in', 'These are some comments 03' );
```

```
INSERT INTO SERVICE ( serv_id, location_id, serv_type_id, serv_catg_id, serv_name,
serv_request_date, serv_complete_date, serv_length, serv_comments )
VALUES ( service_seq.NEXTVAL, '1', '1', '5', 'art photo scanning for L444', '12/18/2003',
'12/18/2003', 'walk-in', 'These are some comments 04' );
```

```
INSERT INTO SERVICE ( serv_id, location_id, serv_type_id, serv_catg_id, serv_name,
serv_request_date, serv_complete_date, serv_length, serv_comments )
VALUES ( service_seq.NEXTVAL, '1', '1', '11', 'help with printing overheads', '01/07/2004',
'01/07/2004', 'walk-in', 'These are some comments 05' );
```

```
INSERT INTO SERVICE ( serv_id, location_id, serv_type_id, serv_catg_id, serv_name,
serv_request_date, serv_complete_date, serv_length, serv_comments )
VALUES ( service_seq.NEXTVAL, '1', '1', '6', 'graphics help with IST newsletter',
'02/15/2004', '02/15/2004', 'walk-in', 'These are some comments 06' );
```

```
INSERT INTO SERVICE ( serv_id, location_id, serv_type_id, serv_catg_id, serv_name,
serv_request_date, serv_complete_date, serv_length, serv_comments )
VALUES ( service_seq.NEXTVAL, '1', '1', '2', 'Community digital video help', '03/10/2004', "",
'ongoing', 'These are some comments 07' );
```

```
INSERT INTO SERVICE ( serv_id, location_id, serv_type_id, serv_catg_id, serv_name,
serv_request_date, serv_complete_date, serv_length, serv_comments )
VALUES ( service_seq.NEXTVAL, '1', '1', '5', 'scanning photos for website', '03/18/2004', "",
'ongoing', 'These are some comments 08' );
```

```
INSERT INTO SERVICE ( serv_id, location_id, serv_type_id, serv_catg_id, serv_name,
serv_request_date, serv_complete_date, serv_length, serv_comments )
VALUES ( service_seq.NEXTVAL, '2', '1', '8', 'help with evaluatiing course', '03/20/2004',
'03/20/2004', 'walk-in', 'These are some comments 09' );
```

```
INSERT INTO SERVICE ( serv_id, location_id, serv_type_id, serv_catg_id, serv_name,
serv_request_date, serv_complete_date, serv_length, serv_comments )
VALUES ( service_seq.NEXTVAL, '3', '1', '7', 'help with L322', '04/05/2004', "", 'walk-in',
'These are some comments 10' );
```

```
--
```

```
-- TABLE: SERV_ASSIGN
```

```
--
```

```
INSERT INTO SERV_ASSIGN ( emp_id, serv_id )  
VALUES ( '1', '1' );
```

```
INSERT INTO SERV_ASSIGN ( emp_id, serv_id )  
VALUES ( '2', '2' );
```

```
INSERT INTO SERV_ASSIGN ( emp_id, serv_id )  
VALUES ( '1', '3' );
```

```
INSERT INTO SERV_ASSIGN ( emp_id, serv_id )  
VALUES ( '3', '4' );
```

```
INSERT INTO SERV_ASSIGN ( emp_id, serv_id )  
VALUES ( '1', '5' );
```

```
INSERT INTO SERV_ASSIGN ( emp_id, serv_id )  
VALUES ( '1', '6' );
```

```
INSERT INTO SERV_ASSIGN ( emp_id, serv_id )  
VALUES ( '2', '7' );
```

```
INSERT INTO SERV_ASSIGN ( emp_id, serv_id )  
VALUES ( '3', '8' );
```

```
INSERT INTO SERV_ASSIGN ( emp_id, serv_id )  
VALUES ( '4', '9' );
```

```
INSERT INTO SERV_ASSIGN ( emp_id, serv_id )  
VALUES ( '1', '10' );
```

```
--
```

```
-- TABLE: SERV_REQUEST
```

```
--
```

```
INSERT INTO SERV_REQUEST ( client_id, serv_id )  
VALUES ( '3', '1' );
```

```
INSERT INTO SERV_REQUEST ( client_id, serv_id )  
VALUES ( '4', '2' );
```

```
INSERT INTO SERV_REQUEST ( client_id, serv_id )  
VALUES ( '1', '3' );
```

```
INSERT INTO SERV_REQUEST ( client_id, serv_id )  
VALUES ( '7', '4' );
```

```
INSERT INTO SERV_REQUEST ( client_id, serv_id )  
VALUES ( '7', '5' );
```

```
INSERT INTO SERV_REQUEST ( client_id, serv_id )  
VALUES ( '1', '6' );
```

```
INSERT INTO SERV_REQUEST ( client_id, serv_id )  
VALUES ( '2', '7' );
```

```
INSERT INTO SERV_REQUEST ( client_id, serv_id )  
VALUES ( '2', '8' );
```

```
INSERT INTO SERV_REQUEST ( client_id, serv_id )  
VALUES ( '5', '9' );
```

```
INSERT INTO SERV_REQUEST ( client_id, serv_id )  
VALUES ( '6', '10' );
```

### **3.0- Creation Spool (table, constraints, sequences) File- /**

Purpose & Method	This section lists the resulting output that was generated when running the various create scripts (tables, constraints, sequences) from section 1.0. Each subsection corresponds and all tables, constraints, and sequences were created successfully.
------------------	---

#### **3.1- Create Table Spool:**

```
SQL> @icats_create_script.sql
```

Table created.

```
SQL> select table_name from user_tables;
```

```
TABLE_NAME
```

```
-----  
CLIENT  
DEPARTMENT  
EMPLOYEE  
EMP_TYPE  
LOCATION  
SERVICE  
SERV_ASSIGN  
SERV_CATEGORY  
SERV_REQUEST  
SERV_TYPE
```

```
10 rows selected.
```

```
SQL> spool off
```

### 3.2- Create Constraints Spool:

```
SQL> edit
```

```
Wrote file afiedt.buf
```

```
1 ALTER TABLE LOCATION ADD(  
2   CONSTRAINT    location_locationid_pk    PRIMARY KEY    (LOCATION_ID)  
3 )
```

SQL> run

```
1 ALTER TABLE LOCATION ADD(  
2   CONSTRAINT    location_locationid_pk    PRIMARY KEY    (LOCATION_ID)  
3 )
```

Table altered.

SQL> edit

Wrote file afiedt.buf

```
1 ALTER TABLE SERV_TYPE ADD(  
2   CONSTRAINT    servtype_servtypeid_pk    PRIMARY KEY  
   (SERV_TYPE_ID)  
3 )
```

SQL> run

```
1 ALTER TABLE SERV_TYPE ADD(  
2   CONSTRAINT    servtype_servtypeid_pk    PRIMARY KEY  
   (SERV_TYPE_ID)  
3 )
```

Table altered.

SQL> edit

Wrote file afiedt.buf

```
1 ALTER TABLE SERV_CATEGORY ADD(  
2   CONSTRAINT    servcatg_servcatgid_pk    PRIMARY KEY  
   (SERV_CATG_ID)  
3 )
```

SQL> run

```
1 ALTER TABLE SERV_CATEGORY ADD(  
2   CONSTRAINT    servcatg_servcatgid_pk    PRIMARY KEY  
   (SERV_CATG_ID)  
3 )
```

Table altered.

SQL> edit

Wrote file afiedt.buf

```
1 ALTER TABLE DEPARTMENT ADD(  
2   CONSTRAINT    department_deptid_pk    PRIMARY KEY    (DEPT_ID)
```

```
3 )
SQL> run
1 ALTER TABLE DEPARTMENT ADD(
2   CONSTRAINT      department_deptid_pk   PRIMARY KEY      (DEPT_ID)
3 )
```

Table altered.

```
SQL> edit
Wrote file afiedt.buf
```

```
1 ALTER TABLE CLIENT ADD(
2   CONSTRAINT      client_clientid_pk PRIMARY KEY (CLIENT_ID),
3   CONSTRAINT      client_deptid_fk   FOREIGN KEY  (DEPT_ID)
REFERENCES DEPARTMENT (DEPT_ID)
4 )
```

```
SQL> run
```

```
1 ALTER TABLE CLIENT ADD(
2   CONSTRAINT      client_clientid_pk PRIMARY KEY (CLIENT_ID),
3   CONSTRAINT      client_deptid_fk   FOREIGN KEY  (DEPT_ID)
REFERENCES DEPARTMENT (DEPT_ID)
4 )
```

Table altered.

```
SQL> edit
Wrote file afiedt.buf
```

```
1 ALTER TABLE EMP_TYPE ADD(
2   CONSTRAINT      emp_type_emp_role_id_pk   PRIMARY KEY
      (EMP_ROLE_ID)
3 )
```

```
SQL> run
```

```
1 ALTER TABLE EMP_TYPE ADD(
2   CONSTRAINT      emp_type_emp_role_id_pk   PRIMARY KEY
      (EMP_ROLE_ID)
3 )
```

Table altered.

```
SQL> edit
Wrote file afiedt.buf
```

```

1 ALTER TABLE EMPLOYEE ADD(
2   CONSTRAINT      employee_emp_id_pk      PRIMARY KEY      (EMP_ID),
3   CONSTRAINT      employee_role_id_fk     FOREIGN KEY      (EMP_ROLE_ID)
REFERENCES EMP_TYPE (EMP_ROLE_ID)
4 )

```

SQL> run

```

1 ALTER TABLE EMPLOYEE ADD(
2   CONSTRAINT      employee_emp_id_pk      PRIMARY KEY      (EMP_ID),
3   CONSTRAINT      employee_role_id_fk     FOREIGN KEY      (EMP_ROLE_ID)
REFERENCES EMP_TYPE (EMP_ROLE_ID)
4 )

```

Table altered.

SQL> edit

Wrote file afiedt.buf

```

1 ALTER TABLE SERVICE ADD(
2   CONSTRAINT      service_servid_pk      PRIMARY KEY      (SERV_ID),
3   CONSTRAINT      service_locationid_fk  FOREIGN KEY      (LOCATION_ID)
REFERENCES LOCATION (LOCATION_ID),
4   CONSTRAINT      service_servtypeid_fk  FOREIGN KEY      (SERV_TYPE_ID)
REFERENCES SERV_TYPE (SERV_TYPE_ID),
5   CONSTRAINT      service_servcatgid_fk  FOREIGN KEY      (SERV_CATG_ID)
REFERENCES SERV_CATEGORY
6 (SERV_CATG_ID)
7* )

```

SQL> run

```

1 ALTER TABLE SERVICE ADD(
2   CONSTRAINT      service_servid_pk      PRIMARY KEY      (SERV_ID),
3   CONSTRAINT      service_locationid_fk  FOREIGN KEY      (LOCATION_ID)
REFERENCES LOCATION (LOCATION_ID),
4   CONSTRAINT      service_servtypeid_fk  FOREIGN KEY      (SERV_TYPE_ID)
REFERENCES SERV_TYPE (SERV_TYPE_ID),
5   CONSTRAINT      service_servcatgid_fk  FOREIGN KEY      (SERV_CATG_ID)
REFERENCES SERV_CATEGORY
6 (SERV_CATG_ID)
7* )

```

Table altered.

SQL> edit

Wrote file afiedt.buf

```

1 ALTER TABLE SERV_ASSIGN ADD(

```

```

2   CONSTRAINT      serv_assign_pk PRIMARY KEY (EMP_ID,SERV_ID),
3   CONSTRAINT      servassign_empid_fk  FOREIGN KEY  (EMP_ID)
REFERENCES EMPLOYEE (EMP_ID),
4   CONSTRAINT      servassign_servid_fk  FOREIGN KEY  (SERV_ID)
REFERENCES SERVICE (SERV_ID)
5* )

```

SQL> run

```

1 ALTER TABLE SERV_ASSIGN ADD(
2   CONSTRAINT      serv_assign_pk PRIMARY KEY (EMP_ID,SERV_ID),
3   CONSTRAINT      servassign_empid_fk  FOREIGN KEY  (EMP_ID)
REFERENCES EMPLOYEE (EMP_ID),
4   CONSTRAINT      servassign_servid_fk  FOREIGN KEY  (SERV_ID)
REFERENCES SERVICE (SERV_ID)
5* )

```

Table altered.

SQL> edit

Wrote file afiedt.buf

```

1 ALTER TABLE SERV_REQUEST ADD(
2   CONSTRAINT      servrequest_pk PRIMARY KEY (CLIENT_ID,SERV_ID),
3   CONSTRAINT      servreq_clientid_fk  FOREIGN KEY  (CLIENT_ID)
REFERENCES CLIENT (CLIENT_ID),
4   CONSTRAINT      servreq_servid_fk  FOREIGN KEY  (SERV_ID)
REFERENCES SERVICE (SERV_ID)
5* )

```

SQL> run

```

1 ALTER TABLE SERV_REQUEST ADD(
2   CONSTRAINT      servrequest_pk PRIMARY KEY (CLIENT_ID,SERV_ID),
3   CONSTRAINT      servreq_clientid_fk  FOREIGN KEY  (CLIENT_ID)
REFERENCES CLIENT (CLIENT_ID),
4   CONSTRAINT      servreq_servid_fk  FOREIGN KEY  (SERV_ID)
REFERENCES SERVICE (SERV_ID)
5* )

```

### 3.3- Create Sequences Spool:

SQL> @icats\_create\_sequence.sql

Sequence created.

Sequence created.

Sequence created.

Sequence created.

Sequence created.

Sequence created.

SQL> spool off

#### **4.0- Table Populate (insert) Spool File- /**

<b>Purpose</b>	This section lists the resulting output that was generated when running the
<b>&amp; Method</b>	table insert/population script from section 2.0. As is indicated, all insert statements were executed successfully

SQL> @insert\_scripts.sql

1 row created.

SQL> spool off

## **5.0- Table Constraint Testing SQL File-** /

Purpose & Method	This section contains the SQL queries used to test our primary and foreign key constraints.
------------------	---

```
--
-- ICATS-- Test Constraints Script
-- Mark Millard and Anna Kaziunas
--

--
-- TABLE: LOCATION
--
INSERT INTO LOCATION ( location_id, location_name, location_size, location_address,
location_phone, location_room, location_building )
VALUES ( 1, 'SOE LTTS Lab', '23', '200 N. Rose Ave.', '856-1111', '2010', 'WWWright Bldg'
)

--
-- TABLE: SERV_TYPE
--
INSERT INTO SERV_TYPE ( serv_type_id, serv_type_name )
VALUES ( 1, 'Project' )

--
-- TABLE: SERV_CATEGORY
--
INSERT INTO SERV_CATEGORY ( serv_catg_id, serv_catg_name )
VALUES ( 1, 'Web Development' );

--
-- TABLE: DEPARTMENT
--
INSERT INTO DEPARTMENT ( dept_id, dept_name )
VALUES ( 'CEP', 'Counseling and Educational Psychology' )

--
-- TABLE: CLIENT
--
```

```
INSERT INTO CLIENT ( client_id, dept_id, client_fname, client_lname, client_username,
client_passwd, client_email, client_phone )
VALUES ( 1, 'GHK', 'Karen', 'Neuman', 'klneuman', '*****', 'klneuman@indiana.edu', '856-
0001' );
```

```
INSERT INTO CLIENT ( client_id, dept_id, client_fname, client_lname, client_username,
client_passwd, client_email, client_phone )
VALUES ( client_seq.NEXTVAL, 'GHK', 'Karen', 'Neuman', 'klneuman', '*****',
'klneuman@indiana.edu', '856-0001' );
```

```
--
```

```
-- TABLE: EMP_TYPE
```

```
--
```

```
INSERT INTO EMP_TYPE ( emp_role_id, emp_role_name )
VALUES ( 'ADM', 'Administrator' );
```

```
--
```

```
-- TABLE: EMPLOYEE
```

```
--
```

```
INSERT INTO EMPLOYEE ( emp_id, emp_fname, emp_lname, emp_username,
emp_passwd, emp_role_id, emp_email, emp_phone )
VALUES ( 1, 'Mark', 'Millard', 'mmillard', '*****', 'CON', 'mmill@indiana.edu', '856-8063' );
```

```
INSERT INTO EMPLOYEE ( emp_id, emp_fname, emp_lname, emp_username,
emp_passwd, emp_role_id, emp_email, emp_phone )
VALUES ( employee_seq.NEXTVAL, 'Mark', 'Millard', 'mmillard', '*****', 'FRTG',
'mmill@indiana.edu', '856-8063' );
```

```
--
```

```
-- TABLE: SERVICE
```

```
--
```

```
INSERT INTO SERVICE ( serv_id, location_id, serv_type_id, serv_catg_id, serv_name,
serv_request_date, serv_complete_date, serv_length, serv_comments )
VALUES ( 1, '1', '1', '1', 'website for P540', '08/10/2003', '08/10/2003', 'walk-in', 'These are
some comments 01' );
```

```
INSERT INTO SERVICE ( serv_id, location_id, serv_type_id, serv_catg_id, serv_name,
serv_request_date, serv_complete_date, serv_length, serv_comments )
VALUES ( service_seq.NEXTVAL, '230', '1', '1', 'website for P540', '08/10/2003',
'08/10/2003', 'walk-in', 'These are some comments 01' );
```

```
INSERT INTO SERVICE ( serv_id, location_id, serv_type_id, serv_catg_id, serv_name,
serv_request_date, serv_complete_date, serv_length, serv_comments )
```

```
VALUES ( service_seq.NEXTVAL, '1', '554', '1', 'website for P540', '08/10/2003',
'08/10/2003', 'walk-in', 'These are some comments 01' );
```

```
INSERT INTO SERVICE ( serv_id, location_id, serv_type_id, serv_catg_id, serv_name,
serv_request_date, serv_complete_date, serv_length, serv_comments )
VALUES ( service_seq.NEXTVAL, '1', '1', '9987', 'website for P540', '08/10/2003',
'08/10/2003', 'walk-in', 'These are some comments 01' );
```

```
--
-- TABLE: SERV_ASSIGN
--
INSERT INTO SERV_ASSIGN ( emp_id, serv_id )
VALUES ( 4322, 4566);
```

```
INSERT INTO SERV_ASSIGN ( emp_id, serv_id )
VALUES ( NULL, NULL);
```

```
--
-- TABLE: SERV_REQUEST
--
INSERT INTO SERV_REQUEST ( client_id, serv_id )
VALUES ( '321', '11222' )
```

```
INSERT INTO SERV_REQUEST ( client_id, serv_id )
VALUES ( NULL, NULL )
```

## **6.0- Table Constraint Test Results Spool File-** /

Purpose & Method	This section lists the resulting spool output that was generated when executing our check constraint SQL queries used to test our primary and foreign key constraints. Errors indicating “unique constraint violated” or “integrity constraint violated” indicates that the PK and FK constraints are working properly.
---------------------	---

```
SQL> @test_constraints.sql
INSERT INTO SERV_CATEGORY ( serv_catg_id, serv_catg_name )
*
ERROR at line 1:
ORA-00001: unique constraint (MMILLARD.SERVCATG_SERVCATGID_PK) violated
```

```
INSERT INTO CLIENT ( client_id, dept_id, client_fname, client_lname, client_username,
client_passwd, client_email, client_phone )
```

\*

ERROR at line 1:

ORA-00001: unique constraint (MMILLARD.CLIENT\_CLIENTID\_PK) violated

```
INSERT INTO CLIENT ( client_id, dept_id, client_fname, client_lname, client_username,
client_passwd, client_email, client_phone )
```

\*

ERROR at line 1:

ORA-02291: integrity constraint (MMILLARD.CLIENT\_DEPTID\_FK) violated - parent key not found

```
INSERT INTO EMP_TYPE ( emp_role_id, emp_role_name )
```

\*

ERROR at line 1:

ORA-00001: unique constraint (MMILLARD.EMP\_TYPE\_EMP\_ROLE\_ID\_PK) violated

```
INSERT INTO EMPLOYEE ( emp_id, emp_fname, emp_lname, emp_username,
emp_passwd, emp_role_id, emp_email, emp_phone )
```

\*

ERROR at line 1:

ORA-00001: unique constraint (MMILLARD.EMPLOYEE\_EMP\_ID\_PK) violated

```
INSERT INTO EMPLOYEE ( emp_id, emp_fname, emp_lname, emp_username,
emp_passwd, emp_role_id, emp_email, emp_phone )
```

\*

ERROR at line 1:

ORA-02291: integrity constraint (MMILLARD.EMPLOYEE\_ROLE\_ID\_FK) violated - parent key not found

```
INSERT INTO SERVICE ( serv_id, location_id, serv_type_id, serv_catg_id, serv_name,
serv_request_date, serv_complete_date, serv_length, serv_comments )
```

\*

ERROR at line 1:

ORA-00001: unique constraint (MMILLARD.SERVICE\_SERVID\_PK) violated

```
INSERT INTO SERVICE ( serv_id, location_id, serv_type_id, serv_catg_id, serv_name,
serv_request_date, serv_complete_date, serv_length, serv_comments )
```

\*

ERROR at line 1:

ORA-02291: integrity constraint (MMILLARD.SERVICE\_LOCATIONID\_FK) violated -

parent key not found

```
INSERT INTO SERVICE ( serv_id, location_id, serv_type_id, serv_catg_id, serv_name,
serv_request_date, serv_complete_date, serv_length, serv_comments )
*
```

ERROR at line 1:

ORA-02291: integrity constraint (MMILLARD.SERVICE\_SERVTYPEID\_FK) violated -  
parent key not found

```
INSERT INTO SERVICE ( serv_id, location_id, serv_type_id, serv_catg_id, serv_name,
serv_request_date, serv_complete_date, serv_length, serv_comments )
*
```

ERROR at line 1:

ORA-02291: integrity constraint (MMILLARD.SERVICE\_SERVCATGID\_FK) violated -  
parent key not found

```
INSERT INTO SERV_ASSIGN ( emp_id, serv_id )
*
```

ERROR at line 1:

ORA-02291: integrity constraint (MMILLARD.SERVASSIGN\_SERVID\_FK) violated -  
parent key not found

```
INSERT INTO SERV_ASSIGN ( emp_id, serv_id )
*
```

ERROR at line 1:

ORA-01400: cannot insert NULL into ("MMILLARD"."SERV\_ASSIGN"."EMP\_ID")

SQL> spool off

## **7.0- Query Generation File (all sql commands)- /**

Purpose & Method	This section contains all of the SQL queries needed for our interface. The queries were determined by looking at the various screens from our interface, and deciding what types of queries would be needed to input or output the proper information. Its important to note that all real values will be determined at the programmatic level from variables, ie) \$clientid, and would be used in the SQL queries instead of hard-coded values, this would add the appropriate dynamic element needed in the queries.
------------------	---

```
--  
-- ICATS-- SQL Queries for Web Application Script  
--  
  
--  
-- SQL for employee login screen (index.html)  
--  
SELECT emp_id, emp_lname, emp_fname, emp_username, emp_passwd FROM  
employee WHERE emp_username like 'mmillard' AND emp_passwd like '*****';  
  
--  
-- SQL for adding client session (record_session.html)  
--  
INSERT INTO SERVICE ( serv_id, location_id, serv_type_id, serv_catg_id, serv_name,  
serv_request_date, serv_complete_date, serv_length, serv_comments )  
VALUES ( service_seq.NEXTVAL, '1', '1', '5', 'scanning photos for website', '04/28/2004', '',  
'walk-in', 'These are some more comments' );  
  
INSERT INTO SERV_REQUEST ( client_id, serv_id )  
VALUES ( 9, service_seq.currval );  
  
  
INSERT INTO SERV_ASSIGN ( emp_id, serv_id )  
VALUES ( 2, service_seq.currval );  
  
  
--  
-- SQL for create report (create_report.html)  
--  
-- SQL for the Client Name query  
SELECT * FROM serv_request, service, client, employee, serv_assign WHERE  
serv_request.client_id = 2  
AND service.serv_id = serv_request.serv_id AND client.client_id = serv_request.client_id  
AND service.serv_id = serv_assign.serv_id AND serv_assign.emp_id = employee.emp_id;  
  
  
-- SQL for the Department query  
SELECT * FROM department, client, serv_request, service, employee, serv_assign  
WHERE client.dept_id = 'CEP' AND department.dept_id = client.dept_id AND client.client_id  
= serv_request.client_id  
AND serv_request.serv_id = service.serv_id AND service.serv_id = serv_assign.serv_id  
AND serv_assign.emp_id = employee.emp_id;  
  
  
-- SQL for the Employee Name query
```

```
SELECT * FROM employee, serv_assign, service, serv_request, client WHERE
employee.emp_id = 1 AND employee.emp_id = serv_assign.emp_id AND
serv_assign.serv_id = service.serv_id AND service.serv_id = serv_request.serv_id AND
serv_request.client_id = client.client_id;
```

-- SQL for the Service Type query

```
SELECT * FROM serv_type, service, serv_assign, employee, serv_request, client WHERE
serv_type.serv_type_id = 1 AND serv_type.serv_type_id = service.serv_type_id AND
service.serv_id = serv_assign.serv_id AND serv_assign.emp_id = employee.emp_id AND
service.serv_id = serv_request.serv_id AND serv_request.client_id = client.client_id;
```

-- SQL for the Service Category query

```
SELECT * FROM serv_category, service, serv_assign, employee, serv_request, client
WHERE serv_category.serv_catg_id = 1 AND serv_category.serv_catg_id =
service.serv_catg_id AND service.serv_id = serv_assign.serv_id AND serv_assign.emp_id
= employee.emp_id AND service.serv_id = serv_request.serv_id AND
serv_request.client_id = client.client_id;
```

-- SQL for the Appointment Type query

```
SELECT * FROM service, serv_assign, employee, serv_request, client WHERE
service.serv_length = 'walk-in' AND service.serv_id = serv_assign.serv_id AND
serv_assign.emp_id = employee.emp_id AND service.serv_id = serv_request.serv_id AND
serv_request.client_id = client.client_id;
```

-- SQL for the Location query

```
SELECT * FROM location, service, serv_assign, employee, serv_request, client WHERE
location.location_id = 3 AND location.location_id = service.location_id AND service.serv_id
= serv_assign.serv_id AND serv_assign.emp_id = employee.emp_id AND service.serv_id =
serv_request.serv_id AND serv_request.client_id = client.client_id;
```

-- SQL for the Request Date query

```
SELECT * FROM service, serv_assign, employee, serv_request, client WHERE
service.serv_request_date = '03/10/2004' AND service.serv_id = serv_assign.serv_id AND
serv_assign.emp_id = employee.emp_id AND service.serv_id = serv_request.serv_id AND
serv_request.client_id = client.client_id;
```

-- SQL for the Completion Date query

```
SELECT * FROM service, serv_assign, employee, serv_request, client WHERE
service.serv_complete_date = '03/10/2004' AND service.serv_id = serv_assign.serv_id AND
serv_assign.emp_id = employee.emp_id AND service.serv_id = serv_request.serv_id AND
serv_request.client_id = client.client_id;
```

```
--OR to check for completion dates that haven't been filled in --
SELECT * FROM service, serv_assign, employee, serv_request, client WHERE
service.serv_complete_date is null AND service.serv_id = serv_assign.serv_id AND
serv_assign.emp_id = employee.emp_id AND service.serv_id = serv_request.serv_id AND
serv_request.client_id = client.client_id;
```

```
--
```

```
-- SQL for Update Employee Information (update_employee.html and subsequent prototype
screens
```

```
--
```

```
-- SQL for the select employee part
```

```
SELECT * FROM employee WHERE emp_id = 1;
```

```
-- SQL for the update employee part
```

```
UPDATE employee SET emp_fname = 'Mark', emp_lname = 'Millard', emp_username =
'mmillard', emp_passwd = '*****', emp_role_id = 'CON', emp_email =
'mmillard@indiana.edu', emp_phone = '856-8063' WHERE emp_id = 1;
```

```
--
```

```
-- SQL for Update Client Information (update_client.html and subsequent prototype
screens(jp.html, etc...))
```

```
--
```

```
-- SQL for the select client part
```

```
SELECT * FROM client WHERE client_id = 3;
```

```
-- SQL for the update client part
```

```
UPDATE client SET dept_id = 'HE', client_fname = 'Barney', client_lname = 'Rubble',
client_username = 'brubble', client_passwd = '*****', client_email = 'barney@indiana.edu',
client_phone = '856-0001' WHERE client_id = 3;
```

```
--
```

```
-- SQL for Add Client (add_client.html)
```

```
--
```

```
INSERT INTO CLIENT ( client_id, dept_id, client_fname, client_lname, client_username,
client_passwd, client_email, client_phone ) VALUES ( client_seq.NEXTVAL, 'IST', 'Anotha',
'Dbuser', 'adbuser', '*****', 'dbuser@indiana.edu', '856-0001' );
```

```
--
```

```
-- SQL for Add Employee (add_employee.html)
```

```
--
INSERT INTO EMPLOYEE ( emp_id, emp_fname, emp_lname, emp_username,
emp_passwd, emp_role_id, emp_email, emp_phone )
VALUES ( employee_seq.NEXTVAL, 'Anotha', 'Employee', 'anemp', '*****', 'CON',
'anemp@indiana.edu', '85-8010'
);
```

## **8.0- Query Results Spool File-** /

Purpose & Method	This section contains the resulting output spool generated by executing all of the queries in section 7.0. In order to simplify the output, the query results for this section were generated with sql*plus using the setting “set mark[up] html on” so that the spool output would be generated in html format, rather than the terribly illegible standard output that sql*plus offers. This method was chosen to provide better human readable output rather than sql*plus default output. Note: the output is in the same order as the query generation file section, but the nature of the wide scrolling tables makes it difficult for printable output. For this reason, we have provided an online version of this section at the URL below.
------------------	--

You may access this query results section at the following URL:

[http://ella.slis.indiana.edu/~mmillard/courses/L546/app\\_query\\_spool.html](http://ella.slis.indiana.edu/~mmillard/courses/L546/app_query_spool.html)