THE SCHOOL OF COMPUTING AND TECHNOLOGY

Calibri, Heading 1, 14pt

EASTERN MEDITERRANEAN UNIVERSITY

201…..-201….. FALL/SPRING SEMESTER

Calibri, Heading 2, 12pt

ITEC404

<<PROJECT TITLE>>
<<PROJECT NUMBER>>

Calibri, 11pt

Prepared By

<<Student Id Name of Member 1>>

<<Student Id Name of Member 2>>

<<…>>

Supervised by

<<Supervisor Name>>

<<Month, Year>>

This is to certify that we have read this project and that in our opinion it is fully adequate, in cope and quality, as an Undergraduate Graduation Project.

<<Supervisor Name>>

<<Graduation Project Committee Chair>>

Acknowledgement Page

Write some words of appreciation to the persons who have supported and made the completion of your graduation project possible. An example can be seen in the following link http://wiki.answers.com/Q/How_do_you_write_an_acknowledgement_in_a_research_report .

Table of Contents

1. Introduction

To assist you in preparing graduation project final report process, the graduation project committee has prepared this Graduation Project Final Report Format Manual. The goal here is to help you prepare a report that looks professional and renders your findings easily accessible to your readers.

2. How to Write Your Report

You may use Microsoft Word or any other equivalent word processor to write your report. It should have an acceptable and readable English level. Note that hand written reports will not be accepted.

3. Organization and Requirements

Reports are organized into three sections:

1. Preliminary Matter
   a) Title Page
   b) Table of Contents
2. Text (Introduction, Current System, Proposed System, System Testing, Implementation Strategy, Maintenance and Support, Outstanding Issues)
3. Back Matter
   a) Appendices (optional)
   b) User's Manual

In the following Sections, each of the above requirements is explained in detail.

3.1 Preliminary Matter: Requirements

Preliminary pages are paginated separately from the rest of the text. Use lower-case Roman numerals at the bottom of these pages. The title page is anticipated as *i*. It is counted but not numbered and should not be included in the Table of Contents. Begin numbering with the Table of Contents using *ii*. Continue using the lower-case Roman numerals up to the first page of Introduction. Specific requirements and examples for each part of the preliminary matter follow.

3.1.1 Title Page

See example in the first page.

3.1.2 Table of Contents

The table of content serves as a map for the project work as it helps you streamline your search for materials. A well written project Table of Content makes it easier for the student to write his or her project work. It also guides the reviewer or supervisor in going through the project work. It should make access easy, not overwhelm the reader with detailed index of the contents. The Table of Contents Page of this manual constitutes an example. The following list of requirements is necessary:

- On a separate page

- Typeface and size: Calibri, 12 point size
- No underlining or italics
- Length: may run more than one page; do not type ''continued'' at the end of the first, or at the beginning of the second page
- Each entry should have tab leaders and corresponding page reference numbers must be aligned correctly.

3.2 Text

The text of the report should be organized logically according to the nature and range of the work being reported. The report must begin with an Introduction, as a separate Chapter, which includes a clear explanation of the goals of the system.

In the section¯Introduction, you should summarize the overview information about the system objectives, scope and constraints. You should also explain the development methodology used, include the project management plan and clearly state the key project benefits towards the stakeholders of the system.

In the section¯Current System, explain the current system environment and functionality. Also mention about the problems and shortcomings relating to the current system.

In the section¯Proposed System, you should explicitly define the proposed functional system using the required UML modeling diagrams and the proposed physical system using the database documentation.

The ¯System Testing is a vital measure taken during and after the implementation of the system in order to evaluate the finished modules/product. You are expected to form a system test strategy, document the overall system test plan and its results.

The¯Implementation Strategy should contain an outline Plan as to whether the existing Data will have to be converted into new format and loaded on new Database structure as well as the most suitable type of System Changeover method required to complete the System Implementation.

The next part¯Maintenance and Support, should discuss if you have any special Maintenance and Support arrangements with the organization.

The final part ¯Outstanding Issues may be brief explanation about any unfinished project tasks due to time limitation or any other hindering factors, any area of the system that requires further investigation etc.

3.2.1 Margins

As noted throughout this manual, you must maintain margins of 2.5 cm on the left, and 2.0 cm at the right, top and bottom of the page. The extra width of the left margin accommodates the binding.

3.2.2 Justification

All text must be right and left justified, just like this manual is written.

3.2.3 Paragraphs and Indentations

The first line of all paragraphs should be indented by one tab space, which provides sufficient guidance to your reader's eye. The indentation must be consistent throughout the text.

3.2.4 Spacing

Use 1.5 line-spacing for the entire text. Materials in tables, appendices, and block quotations, individual footnotes and reference entries may be single-spaced. The Title page must be spaced according to the sample provided. Spacing in the Table of Contents should conform closely to the spacing illustrated in this manual.

3.2.5 Pagination

The text, beginning with the Introduction, or of Chapter 1, should be numbered consecutively. Page numbers must be placed 1.5cm from the bottom center of each page.

3.2.6 Typeface and Size

Calibri typeface with 12 point size should be used throughout the text.

3.2.7. Tables, Figures and Illustrations: General Comments

Tables, figures and illustrations must serve the reader, support your text, and conform to standards in your field. All tables, figures and illustrations must be placed in the Appendix section and referred to within the report using its appendix number. For more information see the Appendix section. While referencing a table, figure, illustration or a series of these within the text, abbreviation as shown in the following examples should be used: A1, A2, A3 etc.

3.2.8. Chapter, Section, and Subsection headings

Chapter, section and subsection headings must all be typewritten in bold, with the following rules:

- Chapter headings should start at a new page, centered, Chapter no in Roman numerals first, followed by the Chapter Title in small letters, the first letters of main words being capital, and with 16 point size Bold Calibri typeface.
- Section headings may start anywhere within the text, after a space of the text of the previous section. Section titles contain Chapter and Section numbers separated by a dot, followed by the Section Title in small letters, the first letters of main words being capital. Section headings should be in bold, 14 point size Bold Calibri typeface.
- Subsection headings should be written similarly as section headings, and contain Chapter number, Section number and Subsection number, separated by dots.

3.3 Back Matter

The back matter of the report consists of the Appendices. The back matter is paginated consecutively from the last page of the text. The back matter, including the appendices, must meet the same margin requirements as the rest of the report.

3.3.1. Appendices

The appendix should contain any tables, figures, illustrations, program listings, diagrams, or other material of technical nature that you have generated for the project. Any document in the appendix must be clearly ordered, and numbered.

Appendices must be designated with a letter (Appendix A, Appendix B, etc.) each starting on a fresh page, and a title. Each appendix must be listed in the Table of Contents. All appendices must meet the usual margin requirements.

Captions for Figures, Tables and Illustrations must be placed at the bottom of each, and centered, as shown in the following example:

A1. This is a Sample Figure

If the table, figure or illustration is too wide for the page, landscape may be used, the caption appearing at the bottom of the table, figure, or illustration (at the right of the page). If the table is too long for one page, the table must be continued on the next page, the title not repeated on the next page, instead, center "Table contd.'' at the top of each succeeding page until the table is completed. Table caption must than follow. If the table is oversized, try to redesign the table to fit the page. If this does not solve the problem, you may reduce the table proportionately, but remember that the captions must still conform to the typeface and size standards of the text.

4. Binding

The final copy must be submitted for binding. The front cover of the binding must be transparent plastic and the black back cover of the binding must be thick hard binding cartoon without spiral.

5. Graduation Report Content Description

The expected sections for the graduation project final report are given in the following sections. The sections with an asterisk (*) are optional and they should be included if it was necessity of the project.

5.1 Introduction

Your report will need an introduction. This should not be more than a couple of paragraph. The introduction is the place for a broad, general view of your material. Avoid details which belong properly to the main sections or appendices.

5.1.1 Project Objectives

List the objectives of the new system. For example:-

- To provide faster and more efficient services to the customers
- To set-up an environment to facilitate the current and future e-commerce system requirements etc.

5.1.2 Project Scope and Constraints

Project Scope means the range or extent of the project. It helps to establish the boundaries of the requested systems since it requires a precise statement of the problem. Along defining the scope of the project, you need to identify any constraints on the system. A constraint is a condition, restriction or requirements that the System must satisfy. A constraint can be in terms of time, money or technical concern.

e.g:- This project must be ready on 1st January 2004
The project cost must not exceed $20,000
The new system must be based on companies existing network infrastructure etc.

5.1.3 Development Methodology

This section should contain the tools and techniques that are used for the Development and implementation of the Proposed System. You should also give a brief justification of the method and technique that you chose for this project. You should consider and justify the usage of the following components.

5.1.3.1 Programming Languages

The chosen Structured Programming Languages or Object-oriented Programming Languages (OOPL).

5.1.3.2 System Architecture and Processing Methods

The system architecture and processing methods such as mainframe architecture, file Server, client/server architecture, web-centric, etc.

5.1.3.3 Database Management Systems

The database management system such as Oracle, SQL-Server, MySQL. Justify your reasons for choosing this particular database management system.

5.1.3.4 Hardware and Communications Platform

The hardware environment and communication platform for which the system should be developed.

5.1.4 Project Development Plan

The Project Plan provides an overall framework for managing the costs and schedules. Project planning takes place at the beginning and end of each Project Phase to develop a plan and schedule for the phase that follows. GANTT chart is a popular project planning and scheduling tool.

5.1.5 Key Project Benefits

Emphasize on the key (major) project benefits. This could include both tangible and intangible benefits that are expected once the system becomes operational. You must justify any benefits you are claiming with proofs in Appendix section.

Example:-

- Significant savings of up to 20% on fleet administration staff costs
- Up to 60% reduction on telephone and telex costs
- Improved customer relationship effect on customer loyalty. (Intangible)

5.2 Current System (Existing System)

A Brief description about the current system, operational environment and its major problems and shortcomings will have to be considered under this heading.

Sufficient details have to be given in order to enable the reader to appreciate and justify the development of proposed new system.

Every organization has an existing system or procedures. The system can be pen-and-paper system otherwise known manual-based system or a computer-based information system.

It is therefore necessary to study and analyze the current system environment, in order to:-

- understand the problems and shortcomings of the current system and generate solutions to solve problems
- Replace the current system with an improved version

Information about the current system can be obtained from several sources by applying fact-finding techniques. One of the main fact-finding technique is User Interviews, other techniques are:- questionnaires, observation, document review, sampling, and research. You may combine and use these techniques to gather information about the current system.

When gathering information about the current system you should focus your attention on the following 5 areas: - system outputs, inputs, processes, timing and system controls. It is also necessary to gather statistical information about the volumes, sizes and frequencies of the current system.

5.3 Proposed System

This section attempts to propose a feasible solution to the problems in the current system environment as well as fulfilling new and/or additional User requirements.

5.3.1 Proposed Functional System

This section will contain details about the main functionality of the Proposed System. In other words it will contain details of Logical Model of the System, in which it will show what the System must do to satisfy business information need rather than how it will be accomplished physically .

5.3.1.1 General use-case Diagram (updated Version)

The general use-case diagram will display all the system functions and all the actors with the relationship lines. The diagram must be updated according to the implemented system functions and actors in ITEC404.

5.3.1.2 Partial use-case Diagrams (updated Version)

The partial use-case diagram will display all the system functions, designed for each actor separately. The diagram must be updated according to the implemented system functions and actors in ITEC404.

5.3.1.3 Use-case Templates (updated Version)

It's a formal method of documenting a use-case in order to understand the business processes and the ways the system will responds to them.

The diagram must be updated according to the implemented system functions and actors used in Graduation Project Implementation Phase (ITEC404).

5.3.1.4 Sequence Diagrams

Sequence diagrams show a detailed flow for a specific use case. They show the calls and message flow between the objects in their sequence and also show the different calls towards different objects. Sequence Diagrams must be in the detailed level showing the actors, user-interface objects, entity objects and controller objects (if used).

5.3.1.5 Package Diagram

Package diagram is **UML** structure diagram which shows packages and dependencies between the packages. **Package** is a namespace used to group together elements that are semantically related and might change together. It is a general purpose mechanism to organize elements into groups to provide better structure for system model and make it simpler and easier to understand.

5.3.2 Proposed Physical System

5.3.2.1 Design Class Diagram

UML design class diagrams show the static structure of the system using software class definitions. Attribute visibility is shown for permanent connections. Classes also include the list of operations (methods).

5.3.2.2 Documentation of Database Implementation

The data modeling task is performed by first defining your entities, the significant objects of interest. Next, you define primary keys for your entities. After you decide on the entities and their primary keys, you can define the relationships that exist between the entities and then you can define additional attributes for the entities. Finally, you normalize the data. The final project report shall contain the logical ERD, relational schema as well as the documentation for all entities, relationships and attributes. The relational schema shall clearly show all the foreign keys and check constraints. All tables are required to be normalized. The report shall **explain** that either the tables are normalized or why some of the tables are un-normalized. The entity, attribute and relationship documents must be completed with the latest information especially with regard to the data types, sizes and etc.

The logical ER-Diagram will use **Crow's Foot Notation** and will be presented in the Appendix section of the report. You may find a sample Logical ER-Diagram in the Appendix A section figure A1**.** It will be followed by the **relational schema** that will clearly show all Primary and Foreign Key constraints. Database documentation will be added to **Appendix A** but must be **discussed and referenced inside the report using its figure number.** Scripts to create database structures (tables, sequences, views, procedures, triggers etc.) and to populate the tables must be discussed and referenced briefly inside the **report** but the codes will be added to the **Appendix A**.

Examples of information that will be added to the appendix can be found in related sections.

5.3.2.3 Database Security and Maintenance Plan

The strategies that will be used to ensure the security of the database (for example, user passwords, profiles, roles, privileges etc.) will be discussed in this section.

Backup and recovery strategies will be discussed in this section as well. For example, what type of backup will be taken according to which schedule, who will be responsible from the backups etc. will be discussed here with appropriate reasoning.

Any relevant script used will be added to **appendix**.

5.4 System Testing

5.4.1 System Test Strategy

Regardless of how the system was developed, the system testing has the following objectives:-

- Perform a final test of all program modules
- Demonstrate that the systems function correctly
- Demonstrate that the system has no errors and defects
- Demonstrate that users can interact with the system successfully
- Verify that all system components are integrated properly and that actual processing situation will be handled correctly.
- Confirm that the system can handle predicted volumes of transaction data in a timely and efficiently manner without crushing etc.

5.4.2 Overall System Test Plan

This section should contain an outline test Plan strategy as well as a detailed Test Plan for each test case. The purpose of the detailed test plan is to demonstrate that the system is tested according to a plan for every known eventuality and that the test results are kept as proofs.

5.4.3 Test Cases

The test results must be included in the Appendix B section of the report with the relevant test plan as proof. For more information about the typical content of a written test case, please check out the following link:

https://en.wikipedia.org/wiki/Test_case

5.5 Implementation Strategy *

This section should contain an outline Plan as to whether the existing Data will have to be converted (data conversion) into new format and loaded on new database structure as well as the most suitable type of system changeover method required to complete the System Implementation. Discuss the problems, if any, you encountered in this phase.

Almost every new system replaces an existing system either a completely manual system or an earlier automated system. Normally the existing system is important and needs to be converted to the format required in the new system. The activity to convert the data often becomes a small project on its own and usually carried out in cooperation with the users.

Although old databases used by the existing systems are commonly reused in the new systems, some changes necessitate changes and reloading of database contents.

System changeover is the process of putting the new system online and retiring the old System. System changeover can be rapid or slow depending on the method. There are four systems changeover methods:-

a) DIRECT CUTOVER - With Direct Cutover, the Old System stops and the New System starts simultaneously. This is the least expensive and most risky method.
b) PARALLEL OPERATION - With this method, both the Old and new systems are operated for some period of time. This is the most expensive and least risky method safest method
c) PILOT OPERATION - A limited part of the organization uses the New System for a period of time while the old system continues to operate for the remaining part of the organization. After the System proves successful at the pilot site it is implemented throughout the organization.
d) PHASED CHANGEOVER – One module of the new System at a time is implemented throughout organization until the entire new System is operational.

Pilot and Parallel changeover methods represent compromise between direct cutover and parallel operation and are less risky than Direct Cutover and less costly than Parallel operation.

5.6 Maintenance and Support

This section will only be present if you have any special maintenance and support arrangements with the organization or if there are any important maintenance issues that you have to pass on the reader of the report.

5.7 Outstanding Issues

This section should contain brief information about the Status of the system. Any concerns and/or suggestions about the project as identified by the original project development team have to be mentioned in this section. The type of information may be brief explanation about any unfinished project tasks due to time limitation or any other hindering factors. If any area of the system that requires further investigation etc.

6. References

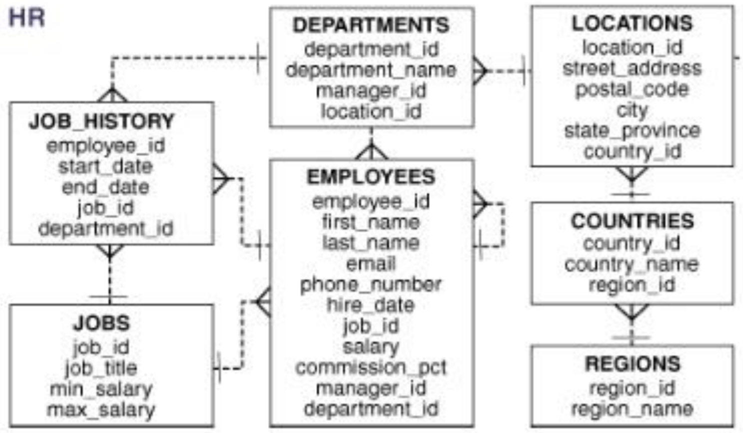
[1] References may include published papers, books and Internet resources.

[2] For published papers, name(s) of the author(s), title of the paper, name of the published journal or proceeding, Volume no, Issue no, ISSN and year published should be written.

[3] For books, name(s) of the author(s), title of the book, publisher name, Edition, ISBN and year published should be written.

[4] For Internet resources, web link for the resource, date visited and date last updated should be written.

Appendix A



A1. Logical ER-Diagram of the proposed system

Entity Name	Description	Aliases	Occurrence
Branch	Place of Work	Office, Branch Office	One or more branches of the company are located in each city throughout the country
Staff	General term describing all staff employed by the company	Employee	Each member of staff works at a particular branch
Secretary	Responsible for supporting secretarial support to staff		Each branch has several secretaries. Each secretary supports a group of staff
Customer	General term describing all customers who make a contract with the company		A customer may have one or more projects
Project	General term describing all projects accepted by the company		A project is requested by a customer from a branch

A2. Entity Type Document

Relationship Type	Entity Type	Cardinality Ratio	Participation
Work For	Staff	M	Mandatory
	Branch	1	Optional
isa-partial	Staff	1	Superclass
	Secretary	M	Subclass
contract	Branch	1	Mandatory
	Project	M	Mandatory
	Customer	1	Mandatory

A3. Relationship Type Document

Entity/Relationship Type	Attribute		Description	Data Type and Length	Constraint	Default Value	Alias	Null ?	Derived ?
Staff	Name	FirstName	First name of the staff	varchar(20)				No	No
		LastName	Last name of the staff	varchar(30)	AK			No	
	id		Uniquely identified staff	number(5)	PK				
	gender		gender of staff	char(1)	CK(‘F’, ‘M’)	‘F’			
Branch	id		Uniquely identifies a branch	number(3)	PK				
	city		City of branch address	varchar2(20)	AK				
Secretary	steno		Shows whether the secretary knows steno or not	char(1)	CK(‘Y’, ‘N’)	Y		No	
	word_min		Typing speed in word per minute	number					
Project	id		Uniquely identifies a project	char(7)	PK				
	title		Name of a project	varchar(100)					
	budget		Budget allocated to a project	number	CK(budget>0)				
Customer	id		Uniquely identifies a customer	number(2)	PK				
	name		Full name of the customer	varchar(50)					
Contract	cont_date		Date the contract is signed	date		current date			
	completed		Shows whether the contract has been completed or not	char(1)	CK(‘Y’, ‘N’)				

A4. Attribute Document

Domain Name	**Domain Characteristics**	**Examples of allowable values**	**Columns**
gender_domain	char(1);(‘M’ or ‘F’)	M, F	Staff(gender)
yes_no_domain	char(1); (‘Y’ or ‘N’)	Y, N	Secretary(steno), Contract(completed)
project_id_domain	char(7); First three characters month abbreviation,4th character underscore, last 3 characters sequential number	JAN_001, JAN_999, APR_067	Project(id)

A5. Attribute Domain Document

```
Rem  Creating EMP_DETAILS_VIEW view ...

CREATE OR REPLACE VIEW emp_details_view
  (employee_id,
   job_id,
   manager_id,
   department_id,
   location_id,
   country_id,
   first_name,
   last_name,
   salary,
   commission_pct,
   department_name,
   job_title,
   city,
   state_province,
   country_name,
   region_name)
AS SELECT
  e.employee_id,
  e.job_id,
  e.manager_id,
  e.department_id,
  d.location_id,
  l.country_id,
  e.first_name,
  e.last_name,
  e.salary,
  e.commission_pct,
  d.department_name,
  j.job_title,
  l.city,
  l.state_province,
  c.country_name,
  r.region_name
FROM
  employees e,
  departments d,
  jobs j,
  locations l,
  countries c,
  regions r
WHERE e.department_id = d.department_id
  AND d.location_id = l.location_id
  AND l.country_id = c.country_id
  AND c.region_id = r.region_id
  AND j.job_id = e.job_id
WITH READ ONLY;
```

```
REM procedure and statement trigger to allow dmls during business hours:
CREATE OR REPLACE PROCEDURE secure_dml
IS
BEGIN
  IF TO_CHAR (SYSDATE, 'HH24:MI') NOT BETWEEN '08:00' AND '18:00'
        OR TO_CHAR (SYSDATE, 'DY') IN ('SAT', 'SUN') THEN
        RAISE_APPLICATION_ERROR (-20205,
               'You may only make changes during normal office hours');
  END IF;
END secure_dml;
/

CREATE OR REPLACE TRIGGER secure_employees
  BEFORE INSERT OR UPDATE OR DELETE ON employees
BEGIN
  secure_dml;
END secure_employees;
/

REM
**************************************************************************
REM procedure to add a row to the JOB_HISTORY table and row trigger
REM to call the procedure when data is updated in the job_id or
REM department_id columns in the EMPLOYEES table:

CREATE OR REPLACE PROCEDURE add_job_history
  (  p_emp_id          job_history.employee_id%type
   , p_start_date      job_history.start_date%type
   , p_end_date        job_history.end_date%type
   , p_job_id          job_history.job_id%type
   , p_department_id   job_history.department_id%type
   )
IS
BEGIN
  INSERT INTO job_history (employee_id, start_date, end_date,
                           job_id, department_id)
    VALUES(p_emp_id, p_start_date, p_end_date, p_job_id, p_department_id);
END add_job_history;
/


CREATE OR REPLACE TRIGGER update_job_history
  AFTER UPDATE OF job_id, department_id ON employees
  FOR EACH ROW
BEGIN
  add_job_history(:old.employee_id, :old.hire_date, sysdate,
                  :old.job_id, :old.department_id);
END;
/

COMMIT;
```

A6. View/Trigger Example

```
-- create tables

CREATE TABLE regions
    ( region_id      NUMBER NOT NULL
    , region_name    VARCHAR2(25)
    );

ALTER TABLE regions
ADD ( CONSTRAINT reg_id_pk
                PRIMARY KEY (region_id)
    ) ;

CREATE TABLE countries
    ( country_id      CHAR(2) NOT NULL
    , country_name    VARCHAR2(40)
    , region_id       NUMBER
    , CONSTRAINT     country_c_id_pk
                    PRIMARY KEY (country_id)
    )

ALTER TABLE countries
ADD ( CONSTRAINT countr_reg_fk
                FOREIGN KEY (region_id)
                 REFERENCES regions(region_id)
    ) ;

CREATE TABLE locations
    ( location_id    NUMBER(4) CONSTRAINT loc_id_pk
                               PRIMARY KEY (location_id)
    , street_address VARCHAR2(40)
    , postal_code    VARCHAR2(12)
    , city           VARCHAR2(30) NOT NULL
    , state_province VARCHAR2(25)
    , country_id     CHAR(2)
    , CONSTRAINT loc_c_id_fk FOREIGN KEY (country_id)
                           REFERENCES countries(country_id)

    ) ;

Rem     Useful for any subsequent addition of rows to locations table
Rem     Starts with 3300

CREATE SEQUENCE locations_seq
 START WITH     3300
 INCREMENT BY   100
 MAXVALUE       9900
 NOCACHE
 NOCYCLE;

CREATE TABLE departments
    ( department_id    NUMBER(4) CONSTRAINT dept_id_pk
                                 PRIMARY KEY (department_id)
    , department_name  VARCHAR2(30)
        CONSTRAINT  dept_name_nn  NOT NULL
    , manager_id       NUMBER(6)
    , location_id      NUMBER(4)
    ) ;


ALTER TABLE departments
ADD (
      CONSTRAINT dept_loc_fk
                FOREIGN KEY (location_id)
                 REFERENCES locations (location_id)
     ) ;
```

```
Rem     Useful for any subsequent addition of rows to departments table
Rem     Starts with 280

CREATE SEQUENCE departments_seq
 START WITH     280
 INCREMENT BY   10
 MAXVALUE       9990
 NOCACHE
 NOCYCLE;


CREATE TABLE jobs
    ( job_id         VARCHAR2(10) CONSTRAINT job_id_pk
                PRIMARY KEY(job_id)
    , job_title      VARCHAR2(35)
       CONSTRAINT     job_title_nn  NOT NULL
    , min_salary     NUMBER(6)
    , max_salary     NUMBER(6)
    ) ;


CREATE TABLE employees
    ( employee_id    NUMBER(6) CONSTRAINT emp_emp_id_pk
                     PRIMARY KEY (employee_id)
    , first_name     VARCHAR2(20)
    , last_name      VARCHAR2(25) NOT NULL
    , email          VARCHAR2(25) NOT NULL
    , phone_number   VARCHAR2(20)
    , hire_date      DATE NOT NULL
    , job_id         VARCHAR2(10) CONSTRAINT emp_job_fk
                     FOREIGN KEY(job_id)REFERENCES jobs (job_id)
                      NOT NULL
    , salary         NUMBER(8,2)
    , commission_pct NUMBER(2,2)
    , manager_id     NUMBER(6)
    , department_id  NUMBER(4) CONSTRAINT emp_dept_fk
                     REFERENCES departments(department_id)
    , CONSTRAINT     emp_salary_min CHECK (salary > 0)
    , CONSTRAINT     emp_email_uk UNIQUE (email)
    ) ;



ALTER TABLE employees
ADD ( CONSTRAINT     emp_manager_fk
                     FOREIGN KEY (manager_id)
                      REFERENCES employees
    ) ;

ALTER TABLE departments
ADD ( CONSTRAINT dept_mgr_fk
                FOREIGN KEY (manager_id)
                 REFERENCES employees (employee_id)
    ) ;



CREATE SEQUENCE employees_seq
 START WITH     207
 INCREMENT BY   1
 NOCACHE
 NOCYCLE;
```

```
CREATE TABLE job_history
    ( employee_id   NUMBER(6)
         CONSTRAINT    jhist_employee_nn  NOT NULL
    , start_date    DATE
        CONSTRAINT    jhist_start_date_nn  NOT NULL
    , end_date      DATE
        CONSTRAINT    jhist_end_date_nn  NOT NULL
    , job_id        VARCHAR2(10)
        CONSTRAINT    jhist_job_nn  NOT NULL
    , department_id NUMBER(4)
    , CONSTRAINT    jhist_date_interval
                    CHECK (end_date > start_date)
    ) ;


ALTER TABLE job_history
ADD ( CONSTRAINT jhist_emp_id_st_date_pk
      PRIMARY KEY (employee_id, start_date)
    , CONSTRAINT     jhist_job_fk
                     FOREIGN KEY (job_id)
                     REFERENCES jobs
    , CONSTRAINT     jhist_emp_fk
                     FOREIGN KEY (employee_id)
                     REFERENCES employees
    , CONSTRAINT     jhist_dept_fk
                     FOREIGN KEY (department_id)
                     REFERENCES departments
    ) ;

COMMIT;

rem   There is a circular foreign key reference between
rem   EMPLOYESS and DEPARTMENTS. That's why we disable
rem   the FK constraints here

ALTER TABLE departments
  DISABLE CONSTRAINT dept_mgr_fk;
-- populate tables
--


INSERT INTO regions VALUES
        ( 1
        , 'Europe'
        );

INSERT INTO regions VALUES
        ( 2
        , 'Americas'
        );

INSERT INTO regions VALUES
        ( 3
        , 'Asia'
        );

INSERT INTO regions VALUES
        ( 4
        , 'Middle East and Africa'
        );


INSERT INTO countries VALUES
        ( 'IT'
        , 'Italy'
        , 1
        );

INSERT INTO countries VALUES
        ( 'JP'
        , 'Japan'
        , 3
        );

INSERT INTO countries VALUES
        ( 'US'
        , 'United States of America'
        , 2
        );

INSERT INTO countries VALUES
        ( 'CA'
        , 'Canada'
        , 2
        );

INSERT INTO countries VALUES
        ( 'CN'
        , 'China'
        , 3
        );

INSERT INTO countries VALUES
        ( 'IN'
        , 'India'
        , 3
        );

INSERT INTO countries VALUES
        ( 'AU'
        , 'Australia'
        , 3
        );

INSERT INTO countries VALUES
        ( 'ZW'
        , 'Zimbabwe'
        , 4
        );

INSERT INTO countries VALUES
        ( 'SG'
        , 'Singapore'
        , 3
        );

INSERT INTO countries VALUES
        ( 'UK'
        , 'United Kingdom'
        , 1
        );

INSERT INTO countries VALUES
        ( 'FR'
        , 'France'
        , 1
        );

INSERT INTO countries VALUES
        ( 'DE'
        , 'Germany'
        , 1
        );

INSERT INTO countries VALUES
        ( 'ZM'
        , 'Zambia'
        , 4
        );

INSERT INTO countries VALUES
        ( 'EG'
        , 'Egypt'
        , 4
        );

INSERT INTO countries VALUES
        ( 'BR'
        , 'Brazil'
        , 2
        );

INSERT INTO countries VALUES
        ( 'CH'
        , 'Switzerland'
        , 1
        );

INSERT INTO countries VALUES
        ( 'NL'
        , 'Netherlands'
        , 1
        );

INSERT INTO countries VALUES
        ( 'MX'
        , 'Mexico'
        , 2
        );

INSERT INTO countries VALUES
        ( 'KW'
        , 'Kuwait'
        , 4
        );

INSERT INTO countries VALUES
        ( 'IL'
        , 'Israel'
        , 4
        );

INSERT INTO countries VALUES
        ( 'DK'
        , 'Denmark'
        , 1
        );

INSERT INTO countries VALUES
        ( 'HK'
        , 'HongKong'
        , 3
        );

INSERT INTO countries VALUES
```

```
        ( 'NG'
        , 'Nigeria'
        , 4
        );

INSERT INTO countries VALUES
        ( 'AR'
        , 'Argentina'
        , 2
        );

INSERT INTO countries VALUES
        ( 'BE'
        , 'Belgium'
        , 1
        );

INSERT INTO locations VALUES
        ( 1000
        , '1297 Via Cola di Rie'
        , '00989'
        , 'Roma'
        , NULL
        , 'IT'
        );

INSERT INTO locations VALUES
        ( 1100
        , '93091 Calle della Testa'
        , '10934'
        , 'Venice'
        , NULL
        , 'IT'
        );

INSERT INTO locations VALUES
        ( 1200
        , '2017 Shinjuku-ku'
        , '1689'
        , 'Tokyo'
        , 'Tokyo Prefecture'
        , 'JP'
        );

INSERT INTO locations VALUES
        ( 1300
        , '9450 Kamiya-cho'
        , '6823'
        , 'Hiroshima'
        , NULL
        , 'JP'
        );

INSERT INTO locations VALUES
        ( 1400
        , '2014 Jabberwocky Rd'
        , '26192'
        , 'Southlake'
        , 'Texas'
        , 'US'
        );

INSERT INTO locations VALUES
        ( 1500
        , '2011 Interiors Blvd'
        , '99236'
        , 'South San Francisco'
        , 'California'
        , 'US'
        );

INSERT INTO locations VALUES
        ( 1600
        , '2007 Zagora St'
        , '50090'
        , 'South Brunswick'
        , 'New Jersey'
        , 'US'
        );

INSERT INTO locations VALUES
        ( 1700
        , '2004 Charade Rd'
        , '98199'
        , 'Seattle'
        , 'Washington'
        , 'US'
        );

INSERT INTO locations VALUES
        ( 1800
        , '147 Spadina Ave'
        , 'M5V 2L7'
        , 'Toronto'
        , 'Ontario'
        , 'CA'
        );

INSERT INTO locations VALUES
        ( 1900
        , '6092 Boxwood St'
        , 'YSW 9T2'
        , 'Whitehorse'
        , 'Yukon'
        , 'CA'
        );

INSERT INTO locations VALUES
        ( 2000
        , '40-5-12 Laogianggen'
        , '190518'
        , 'Beijing'
        , NULL
        , 'CN'
        );

INSERT INTO locations VALUES
        ( 2100
        , '1298 Vileparle (E)'
        , '490231'
        , 'Bombay'
        , 'Maharashtra'
        , 'IN'
        );

INSERT INTO locations VALUES
        ( 2200
        , '12-98 Victoria Street'
        , '2901'
        , 'Sydney'
        , 'New South Wales'
        , 'AU'
        );

INSERT INTO locations VALUES
        ( 2300
        , '198 Clementi North'
        , '540198'
        , 'Singapore'
        , NULL
        , 'SG'
        );


INSERT INTO departments VALUES
        ( 10
        , 'Administration'
        , 200
        , 1700
        );

INSERT INTO departments VALUES
        ( 20
        , 'Marketing'
        , 201
        , 1800
        );

INSERT INTO departments VALUES
        ( 30
        , 'Purchasing'
        , 114
        , 1700
       );

INSERT INTO departments VALUES
        ( 40
        , 'Human Resources'
        , 203
        , 2400
        );

INSERT INTO departments VALUES
        ( 50
        , 'Shipping'
        , 121
        , 1500
        );

INSERT INTO departments VALUES
        ( 60
        , 'IT'
        , 103
        , 1400
        );

INSERT INTO departments VALUES
        ( 70
        , 'Public Relations'
        , 204
        , 2700
        );

INSERT INTO departments VALUES
        ( 80
        , 'Sales'
        , 145
        , 2500
        );

INSERT INTO departments VALUES
        ( 90
        , 'Executive'
        , 100
        , 1700
        );

INSERT INTO jobs VALUES
        ( 'AD_PRES'
        , 'President'
        , 20000
        , 40000
        );
INSERT INTO jobs VALUES
        ( 'AD_VP'
        , 'Administration Vice President'
        , 15000
        , 30000
        );

INSERT INTO jobs VALUES
        ( 'AD_ASST'
        , 'Administration Assistant'
        , 3000
        , 6000
        );

INSERT INTO jobs VALUES
        ( 'FI_MGR'
        , 'Finance Manager'
        , 8200
        , 16000
        );

INSERT INTO jobs VALUES
        ( 'FI_ACCOUNT'
        , 'Accountant'
        , 4200
        , 9000
        );

INSERT INTO employees VALUES
        ( 100
        , 'Steven'
        , 'King'
        , 'SKING'
        , '515.123.4567'
        , TO_DATE('17-JUN-1987', 'dd-MON-yyyy')
        , 'AD_PRES'
        , 24000
        , NULL
        , NULL
        , 90
        );

INSERT INTO employees VALUES
        ( 101
        , 'Neena'
        , 'Kochhar'
        , 'NKOCHHAR'
        , '515.123.4568'
        , TO_DATE('21-SEP-1989', 'dd-MON-yyyy')
        , 'AD_VP'
        , 17000
        , NULL
        , 100
        , 90
        );

INSERT INTO employees VALUES
        ( 102
        , 'Lex'
        , 'De Haan'
        , 'LDEHAAN'
        , '515.123.4569'
        , TO_DATE('13-JAN-1993', 'dd-MON-yyyy')
        , 'AD_VP'
        , 17000
        , NULL
        , 100
        , 90
        );


INSERT INTO employees VALUES
        ( 103
        , 'Alexander'
        , 'Hunold'
        , 'AHUNOLD'
        , '590.423.4567'
        , TO_DATE('03-JAN-1990', 'dd-MON-yyyy')
        , 'IT_PROG'
        , 9000
        , NULL
        , 102
        , 60
        );

INSERT INTO employees VALUES
        ( 104
        , 'Bruce'
        , 'Ernst'
        , 'BERNST'
        , '590.423.4568'
        , TO_DATE('21-MAY-1991', 'dd-MON-yyyy')
        , 'IT_PROG'
        , 6000
        , NULL
        , 103
        , 60
        );

INSERT INTO employees VALUES
        ( 105
        , 'David'
        , 'Austin'
        , 'DAUSTIN'
        , '590.423.4569'
        , TO_DATE('25-JUN-1997', 'dd-MON-yyyy')
        , 'IT_PROG'
        , 4800
        , NULL
        , 103
        , 60
        );
```

```
INSERT INTO job_history
VALUES (102
       , TO_DATE('13-JAN-1993', 'dd-MON-yyyy')
       , TO_DATE('24-JUL-1998', 'dd-MON-yyyy')
       , 'IT_PROG'
       , 60);

INSERT INTO job_history
VALUES (101
       , TO_DATE('21-SEP-1989', 'dd-MON-yyyy')
       , TO_DATE('27-OCT-1993', 'dd-MON-yyyy')
       , 'AC_ACCOUNT'
       , 110);


INSERT INTO job_history
VALUES (101
       , TO_DATE('28-OCT-1993', 'dd-MON-yyyy')
       , TO_DATE('15-MAR-1997', 'dd-MON-yyyy')
       , 'AC_MGR'
       , 110);

INSERT INTO job_history
VALUES (201
       , TO_DATE('17-FEB-1996', 'dd-MON-yyyy')
       , TO_DATE('19-DEC-1999', 'dd-MON-yyyy')
       , 'MK_REP'
       , 20);

INSERT INTO job_history
VALUES  (114
        , TO_DATE('24-MAR-1998', 'dd-MON-yyyy')
        , TO_DATE('31-DEC-1999', 'dd-MON-yyyy')
        , 'ST_CLERK'
        , 50
        );

INSERT INTO job_history
VALUES  (122
        , TO_DATE('01-JAN-1999', 'dd-MON-yyyy')
        , TO_DATE('31-DEC-1999', 'dd-MON-yyyy')
        , 'ST_CLERK'
        , 50
        );

INSERT INTO job_history
VALUES  (200
        , TO_DATE('17-SEP-1987', 'dd-MON-yyyy')
        , TO_DATE('17-JUN-1993', 'dd-MON-yyyy')
        , 'AD_ASST'
        , 90
        );



ALTER TABLE departments
  ENABLE CONSTRAINT dept_mgr_fk;

COMMIT;
```

```
-- populate tables

rem   There is a circular foreign key reference between
rem   EMPLOYESS and DEPARTMENTS. That's why we disable
rem   the FK constraints here

ALTER TABLE departments
  DISABLE CONSTRAINT dept_mgr_fk;


INSERT INTO regions VALUES
        ( 1
        , 'Europe'
        );

INSERT INTO regions VALUES
        ( 2
        , 'Americas'
        );

INSERT INTO regions VALUES
        ( 3
        , 'Asia'
        );

INSERT INTO regions VALUES
        ( 4
        , 'Middle East and Africa'
        );


INSERT INTO countries VALUES
        ( 'IT'
        , 'Italy'
        , 1
        );

INSERT INTO countries VALUES
        ( 'JP'
        , 'Japan'
       , 3
        );

INSERT INTO countries VALUES
        ( 'US'
        , 'United States of America'
        , 2
        );

INSERT INTO countries VALUES
        ( 'CA'
        , 'Canada'
        , 2
        );

INSERT INTO countries VALUES
        ( 'CN'
        , 'China'
        , 3
        );

INSERT INTO countries VALUES
        ( 'IN'
        , 'India'
        , 3
        );

INSERT INTO countries VALUES
        ( 'AU'
        , 'Australia'
        , 3
        );

INSERT INTO countries VALUES
        ( 'ZW'
        , 'Zimbabwe'
        , 4
        );

INSERT INTO countries VALUES
        ( 'SG'
        , 'Singapore'
        , 3
        );

INSERT INTO countries VALUES
        ( 'UK'
        , 'United Kingdom'
        , 1
        );

INSERT INTO countries VALUES
        ( 'FR'
        , 'France'
        , 1
        );

INSERT INTO countries VALUES
        ( 'DE'
        , 'Germany'
        , 1
        );

INSERT INTO countries VALUES
        ( 'ZM'
        , 'Zambia'
        , 4
        );

INSERT INTO countries VALUES
        ( 'EG'
        , 'Egypt'
        , 4
        );

INSERT INTO countries VALUES
        ( 'BR'
        , 'Brazil'
        , 2
        );

INSERT INTO countries VALUES
        ( 'CH'
        , 'Switzerland'
        , 1
        );

INSERT INTO countries VALUES
        ( 'NL'
        , 'Netherlands'
        , 1
        );

INSERT INTO countries VALUES
        ( 'MX'
        , 'Mexico'
        , 2
        );

INSERT INTO countries VALUES
        ( 'KW'
        , 'Kuwait'
        , 4
        );

INSERT INTO countries VALUES
        ( 'IL'
        , 'Israel'
        , 4
        );

INSERT INTO countries VALUES
        ( 'DK'
        , 'Denmark'
        , 1
        );

INSERT INTO countries VALUES
        ( 'HK'
        , 'HongKong'
        , 3
        );

INSERT INTO countries VALUES
        ( 'NG'
        , 'Nigeria'
        , 4
        );

INSERT INTO countries VALUES
        ( 'AR'
        , 'Argentina'
        , 2
        );

INSERT INTO countries VALUES
        ( 'BE'
        , 'Belgium'
        , 1
        );

INSERT INTO locations VALUES
        ( 1000
        , '1297 Via Cola di Rie'
        , '00989'
        , 'Roma'
        , NULL
        , 'IT'
        );

INSERT INTO locations VALUES
        ( 1100
        , '93091 Calle della Testa'
        , '10934'
        , 'Venice'
        , NULL
        , 'IT'
        );

INSERT INTO locations VALUES
        ( 1200
        , '2017 Shinjuku-ku'
        , '1689'
        , 'Tokyo'
        , 'Tokyo Prefecture'
        , 'JP'
        );

INSERT INTO locations VALUES
        ( 1300
        , '9450 Kamiya-cho'
        , '6823'
        , 'Hiroshima'
        , NULL
        , 'JP'
        );

INSERT INTO locations VALUES
        ( 1400
        , '2014 Jabberwocky Rd'
        , '26192'
        , 'Southlake'
        , 'Texas'
        , 'US'
        );

INSERT INTO locations VALUES
        ( 1500
        , '2011 Interiors Blvd'
        , '99236'
        , 'South San Francisco'
        , 'California'
        , 'US'
        );

INSERT INTO locations VALUES
        ( 1600
        , '2007 Zagora St'
        , '50090'
        , 'South Brunswick'
        , 'New Jersey'
        , 'US'
        );

INSERT INTO locations VALUES
        ( 1700
        , '2004 Charade Rd'
        , '98199'
        , 'Seattle'
        , 'Washington'
        , 'US'
        );

INSERT INTO locations VALUES
        ( 1800
        , '147 Spadina Ave'
        , 'M5V 2L7'
        , 'Toronto'
        , 'Ontario'
        , 'CA'
        );

INSERT INTO locations VALUES
        ( 1900
        , '6092 Boxwood St'
        , 'YSW 9T2'
        , 'Whitehorse'
        , 'Yukon'
        , 'CA'
        );

INSERT INTO locations VALUES
        ( 2000
        , '40-5-12 Laogianggen'
        , '190518'
        , 'Beijing'
        , NULL
        , 'CN'
        );

INSERT INTO locations VALUES
        ( 2100
        , '1298 Vileparle (E)'
        , '490231'
        , 'Bombay'
        , 'Maharashtra'
        , 'IN'
        );

INSERT INTO locations VALUES
        ( 2200
        , '12-98 Victoria Street'
        , '2901'
        , 'Sydney'
        , 'New South Wales'
        , 'AU'
        );

INSERT INTO locations VALUES
        ( 2300
        , '198 Clementi North'
        , '540198'
        , 'Singapore'
        , NULL
        , 'SG'
        );


INSERT INTO departments VALUES
        ( 10
        , 'Administration'
        , 200
        , 1700
        );

INSERT INTO departments VALUES
        ( 20
        , 'Marketing'
        , 201
        , 1800
        );

INSERT INTO departments VALUES
        ( 30
        , 'Purchasing'
        , 114
        , 1700
        );

INSERT INTO departments VALUES
        ( 40
        , 'Human Resources'
        , 203
        , 2400
        );

INSERT INTO departments VALUES
        ( 50
        , 'Shipping'
        , 121
        , 1500
        );

INSERT INTO departments VALUES
        ( 60
        , 'IT'
        , 103
        , 1400
        );

INSERT INTO departments VALUES
        ( 70
        , 'Public Relations'
        , 204
        , 2700
        );

INSERT INTO departments VALUES
        ( 80
        , 'Sales'
        , 145
        , 2500
        );

INSERT INTO departments VALUES
        ( 90
        , 'Executive'
        , 100
        , 1700
        );

INSERT INTO jobs VALUES
        ( 'AD_PRES'
        , 'President'
        , 20000
        , 40000
        );
INSERT INTO jobs VALUES
        ( 'AD_VP'
        , 'Administration Vice President'
        , 15000
        , 30000
        );

INSERT INTO jobs VALUES
        ( 'AD_ASST'
        , 'Administration Assistant'
        , 3000
        , 6000
        );
```

```
INSERT INTO jobs VALUES
        ( 'FI_MGR'
        , 'Finance Manager'
        , 8200
        , 16000
        );

INSERT INTO jobs VALUES
        ( 'FI_ACCOUNT'
        , 'Accountant'
        , 4200
        , 9000
        );

INSERT INTO employees VALUES
        ( 100
        , 'Steven'
        , 'King'
        , 'SKING'
        , '515.123.4567'
        , TO_DATE('17-JUN-1987', 'dd-MON-yyyy')
        , 'AD_PRES'
        , 24000
        , NULL
        , NULL
        , 90
        );

INSERT INTO employees VALUES
        ( 101
        , 'Neena'
        , 'Kochhar'
        , 'NKOCHHAR'
        , '515.123.4568'
        , TO_DATE('21-SEP-1989', 'dd-MON-yyyy')
        , 'AD_VP'
        , 17000
        , NULL
        , 100
        , 90
        );

INSERT INTO employees VALUES
        ( 102
        , 'Lex'
        , 'De Haan'
        , 'LDEHAAN'
        , '515.123.4569'
        , TO_DATE('13-JAN-1993', 'dd-MON-yyyy')
        , 'AD_VP'
        , 17000
        , NULL
        , 100
        , 90
        );


INSERT INTO employees VALUES
        ( 103
        , 'Alexander'
        , 'Hunold'
        , 'AHUNOLD'
        , '590.423.4567'
        , TO_DATE('03-JAN-1990', 'dd-MON-yyyy')
        , 'IT_PROG'
        , 9000
        , NULL
        , 102
        , 60
        );

INSERT INTO employees VALUES
        ( 104
        , 'Bruce'
        , 'Ernst'
        , 'BERNST'
        , '590.423.4568'
        , TO_DATE('21-MAY-1991', 'dd-MON-yyyy')
        , 'IT_PROG'
        , 6000
        , NULL
        , 103
        , 60
        );

INSERT INTO employees VALUES
        ( 105
        , 'David'
        , 'Austin'
        , 'DAUSTIN'
        , '590.423.4569'
        , TO_DATE('25-JUN-1997', 'dd-MON-yyyy')
        , 'IT_PROG'
        , 4800
        , NULL
        , 103
        , 60
        );

INSERT INTO job_history
VALUES (102
       , TO_DATE('13-JAN-1993', 'dd-MON-yyyy')
       , TO_DATE('24-JUL-1998', 'dd-MON-yyyy')
       , 'IT_PROG'
       , 60);

INSERT INTO job_history
VALUES (101
       , TO_DATE('21-SEP-1989', 'dd-MON-yyyy')
       , TO_DATE('27-OCT-1993', 'dd-MON-yyyy')
       , 'AC_ACCOUNT'
       , 110);


INSERT INTO job_history
VALUES (101
       , TO_DATE('28-OCT-1993', 'dd-MON-yyyy')
       , TO_DATE('15-MAR-1997', 'dd-MON-yyyy')
       , 'AC_MGR'
       , 110);

INSERT INTO job_history
VALUES (201
       , TO_DATE('17-FEB-1996', 'dd-MON-yyyy')
       , TO_DATE('19-DEC-1999', 'dd-MON-yyyy')
       , 'MK_REP'
       , 20);

INSERT INTO job_history
```

```
VALUES  (114
        , TO_DATE('24-MAR-1998', 'dd-MON-yyyy')
        , TO_DATE('31-DEC-1999', 'dd-MON-yyyy')
        , 'ST_CLERK'
        , 50
        );

INSERT INTO job_history
VALUES  (122
        , TO_DATE('01-JAN-1999', 'dd-MON-yyyy')
        , TO_DATE('31-DEC-1999', 'dd-MON-yyyy')
        , 'ST_CLERK'
        , 50
        );

INSERT INTO job_history
VALUES  (200
        , TO_DATE('17-SEP-1987', 'dd-MON-yyyy')
        , TO_DATE('17-JUN-1993', 'dd-MON-yyyy')
        , 'AD_ASST'
        , 90
        );



ALTER TABLE departments
  ENABLE CONSTRAINT dept_mgr_fk;

COMMIT;
```

A8. Table Creation and Population Example

Appendix B: Test Cases

For more information about the typical content of a written test case, please check out the following link:

https://en.wikipedia.org/wiki/Test_case

Appendix C:
User's Manual

USER'S MANUAL consists of written and/or other visual information about an application system, as to how it works and how to use it. This section should be organized as a separate report with its own headings and sub-headings.

Note:

- The User's Manual can be compiled from your ¯HELP menu on the Screen.

- Follow the guidelines in the link below to prepare the user's manual http://www.ehow.com/how_4827020_write-user-manual-software.html
- An example user's manual is provided in the link below: http://www.progress.com/Services/renew_online_ug.pdf
- Besides the printed copy, the soft copy of the user's manual should be provided in PDF format.