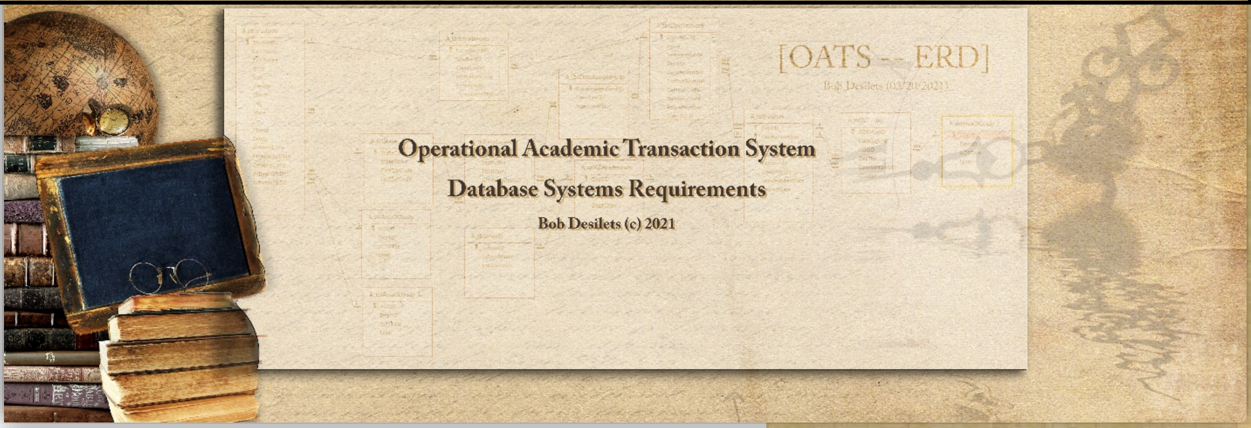
2021

CIS 228 – Operational Academic Transaction System[B] [Fall 2021]


[OATS -- ERD]
Bob Desilets (03/20/2021)
Operational Academic Transaction System
Database Systems Requirements
Bob Desilets (c) 2021

Bob Desilets ©2021

10/29/2021

Contents

Project Case Study Background Information

Mission Statement

Wheelwright Community College (WCC)'s mission is to offer high quality and affordable education which meet the educational and community needs in central Massachusetts. WCC strives to provide excellence in academics through both traditional degree and certification programs which can lead to career development, transfer to a four-year degree, entry into in-demand occupational careers and other learning opportunities.

The Academic Operational Business Environment

The Academic Operational Business Environment is an organization of employees, the business, and students, the clientele. An academic institution provides classes, professors, scheduling, advisors, and other services for students to achieve their goals. These services are paid for by the students using a variety of sources including financial aid, scholarships and more.

Wheelwright Community College (WCC) in particular is a public commuter academic institution located in Worcester, MA. The over 700 employees made up of faculty, administration, and staff at WCC has led to its success, as well as that of its students. It offers over 100 associate degrees and certificate programs with flexible class schedules so that students may readily join the workforce, enhance their job skills, or transfer for a 4-year degree. They can be completed in one or two years. The courses are taught by hands-on experienced faculty. Fields of study include Healthcare, Liberal Arts, Education and more. Over 10,000 students are enrolled at WCC.

One of the most popular fields of study at WCC is Computer and Information Technology, which includes associate degrees and certificate programs for Computer and Information Systems (CIS) and Computer Systems Engineering Technology (CSET), both with specialized options. Students can concentrate their studies in web development, databases, applications, cybersecurity, forensics, and others.

Wheelwright Community College helps all students navigate the costs that higher education requires. Our Financial Aid Office staff communicates with students and their advisors by email, phone, and by appointment during COVID-19 restrictions. We assist students from the first day of enrollment to graduation. WCC provides step by step help applying for all loans. The financial aid options include scholarships and grants from both state and

federal programs. Requiring WCC access to state and federal applications to connect students with the appropriate funding. WCC also has a variety of institutional financial assistance programs. There are extensive services available to Veterans and those with Disabilities.

Essential Functions

An academic operational database system is a crucial tool that may be used by several users with different needs at WCC, including staff, administration, professors, and students. It can track the important day-to-day operations that determine the college’sbusiness and academic goals.

The database performs the essential functions of data manipulation:

- Create Data
- Record Data
- Update Data
- Delete Data

The above four functions are collectedly known as CRUD.

For staff, some data manipulation operations and queries needed are for:

- Demographic Information, particularly for marketing, grants, financial aid and more
- Degree/Program Information
- Financial Aid Status
- Student Status, such as Academic Probation, Dean’s List
- Tuitions

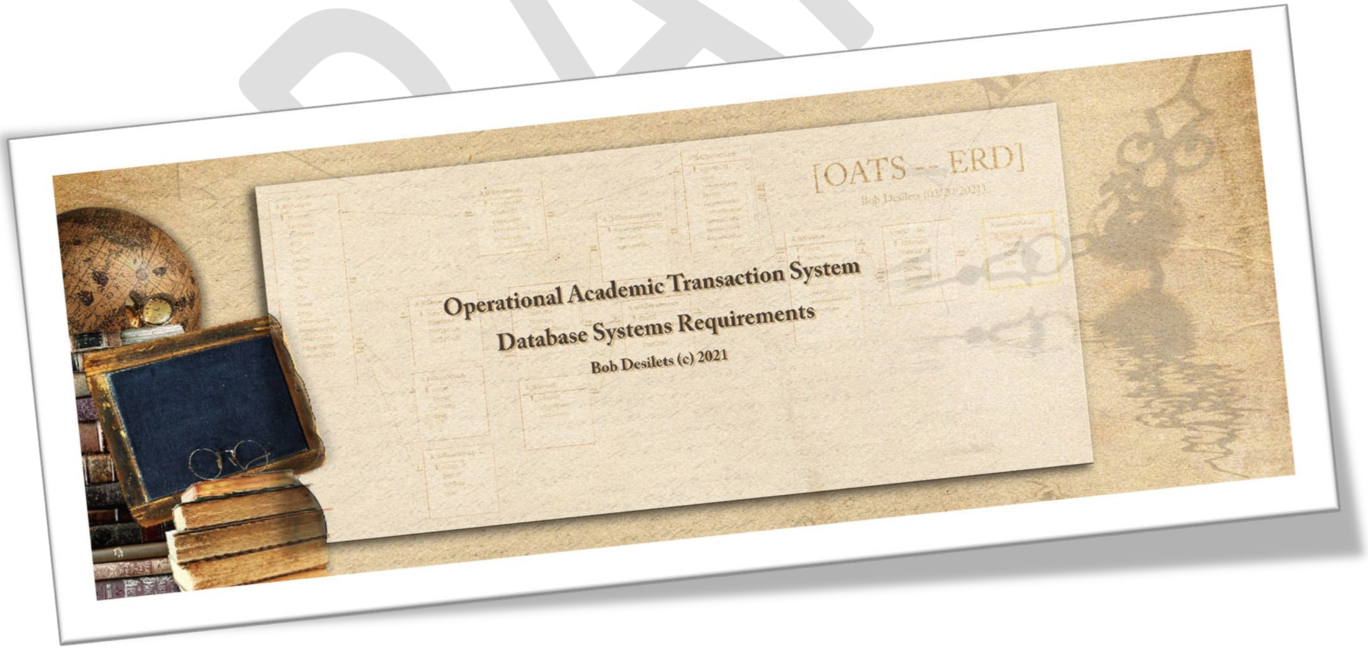
For administration and professors, the following are needed:

- Academic Records
- Advisee Information
- Applicant Records
- Class Scheduling
- Financial Aid Inquiries
- Grade Reports
- Pre-registration Testing (Math, English)

OATS COURSE PROJECT INTRODUCTION/REQUIREMENTS

The goal of this project to is develop a data model for an Academic institution to:

- Help track their students
- Help track their classes
- Help collect data for marketing open classes and areas of study (degrees/certificates)
- Help research student’s requirement needed for advising and program completion
- Help forecast the number of seats required for current demands.
- Keep track of students’ progress.
- Provide advising with required information to advise their advisees.
- OTHERS…


[OATS – ERD]
Operational Academic Transaction System
Database Systems Requirements
Bob Desilets (c) 2021

Project Introduction

The [OATS] Operational Academic Transaction System is an application with a basic aim of creating an online environment to provide easy and flexible assistance to the user of any education institution to apply the CRUD (Create, Retrieve, Update, & Delete) processes to their institutional data. This also includes the ability to print details of its students and staff. The OATS project for this class will focus on the backend processes. We shall use Oracle SQL to implement the CRUD processes required to:

1. Create the data structure.
2. Insert, Update and Delete data to this database structure.
3. Select required row(s) and table(s) to build simple and complex queries for data extraction.

Thus, providing a dynamic platform to execute its data processing online or offline. This Application/Data structure provides a single source of data repository for streamlining your Institution's business processes and reporting purposes.

Make sure to attach "ALL" work: Project Reference Guide ".DOCX", Project presentation ".PPTX" (use PPT Notes) ... And any other required materials... Make sure to hand Bob hardcopy of all work at the beginning of the final exam period... Make sure to insert footnotes in your documentation to show placement of required project components such as:
A-6 - [Class Rosters Grade Sheet]

Make sure to use the project specifications as your check list!

PLEASE let me know if you have any questions.

In order to receive any credit for this Course Project Assignment you must be at our final Bb Collaborate meeting (Monday December 20th at 2PM Bb Collaboration. Be early.) We will be starting right at 4 PM... Be early with all your work submitted. Be ready to present! Submittal closes right at 2 PM.

1. Submit your project electronic copy through Blackboard

2. Present your project to the class

Course project requirements

Final exam period Monday 12/20/2021 [2 – 4 PM]

- Follow the attached .PDF for project specification...

Make sure to attach "ALL" work: Project Reference Guide ".DOCX", Project presentation ".PPTX" (use PPT Notes) You must also include your PPT presentation as an appendix of your Project Reference Guide ... And any other required materials... Make sure to insert footnotes in your documentation to show placement of required project components such as:
If Missing A1, A2, A3, A4, … -10 will be deducted for each one missing

Make sure that you are ready to present your presentation right at 2 PM on Bb Collaborate!!!

Report must include:

- Cover page
- an executive summary
- Table of content (Must use the Word built-in feature
- page numbers on each page
- Footnotes to make reference for each required element
    - Sample (1) A4 - [Non-Declared Student List for students that live in Worcester]
- Appendixes with ALL THE REQUIRED SCRIPTS. This should include all provide scripts and your own work.
- Appendix with other supporting materials.

Have your presentation ready to present

Also 25% will be deducted if it is submitted to Blackboard after 1 PM

Make sure to have all your project material uploaded to Blackboard before the start of the exam period! Do not zip your deliverables. Make sure to attach two separate files (.DOCX and PPTX)

Project Requirements

Your task is to produce the SQL code required to[1]:

A. Project Specifications

1. Create, test, and document all required SQL statement for all the necessary database objects. Run the supplied load script to build and populate the OATS Database:
   - a) Step 1 – Create (Define the OATS data structure)
   - b) Step 2 – Insert (Add data to tables)
   - c) Step 3 – Alter (Define table constraints)
   - d) Step 4 – Several Zip Codes Rutland, Ma. Are not correct. [01543 is the correct Zip Code.]
     1. Examine the tables that use the Zip Code field and identify which Student's/Employee's Zip Code requires changing. no incorrect zip codes where city = Rutland?
     2. Update the rows requiring modification. Use the SQL UPDATE statement is used to update existing records in the tables [01543 is the correct Zip Code.]

2. Add/Create A_tblSemesterCodes[2] B_tblSemesterCodes
   - a) [PK] Semester_Code [2-digit integer]
   - b) Semester_Title [50 text character]
   - c) Add|Modify a foreign key Semester_Code to A_ClassSections linkinking to A_tblSemesterCodes gotta do D) before C)
   - d) INSERT Data: [10|Summer I, 20|Summer II, 30|Fall, 40|Intersession, 50|Spring] Please note that the year throughout the data structure indicates the academic year. Therefore, Year-2015 | SemesterCode-50 translate to Academic year 2015-2016 Spring (2016). typed a script and added it rather then running the lines of code

3. [Student List with active Area of Studies][3] (Just check for the first AOS)
   active meaning not undecided?

[1] Note that this is an ill define list of tasks. It is intended to provide you with the utmost space for creativity within a realist project scope.

[2] Chapters 3, 4 & 5

[3] Chapter 8 Order by

[Act_AOS_ID1, StudentID, LastName, FirstName] sort by (Act_AOS_ID1 then StudentID)]

4. [Non-Declared Student List for students that live in Worcester][4] (Check the zipcode field using the "LIKE" statement "016" then ACT_AOS_ID1 for NULLS or 'UNDC')

   act_aos_id1 is not nullable???

   [[LastName, FirstName, StudentID, Act_AOS_ID1, Street, City, State_Code, Zip_Code] sorted by (LastName, FirstName)

a. find hanson
b.find no advisor
c.update advisor
d. run advisors

5. [Update AOS('CIWP') students (A_tablStudents) without a current advisor to an advisor of 'Russell Hanson')][5] Hint: Find Professor Hanson's ID in the Employee table. Display both the before and after data (Use the SELECT statement).

   aos1 and 2?

6. [Class Rosters Grade Sheet][6] for a Specific Class Section (You will need to research to find the Class_SectID for "CIS 121-40" Fall 2015…] List for a specific class section all its students Hint: requires joining multiple tables. Locate what tables contain the required data columns.

   (Year, SemesterTitle, DeptID, CourseNumber, CourseTitle, LastName, FirstName, StudentID, FianlGrade) sort by Student's Last and First Names

7. [Display Employee Individual Zip Code Count for car pool study][7] Hint: Use the Group by.

   (Zip_Code, Count) sort by Zip_Code

```
SELECT employeeid, city, zip_code
   FROM b_tblemployees
   WHERE zip_code = '01420';
checked on count using the above
```

8. [Fall 2016 Needs Registration][8] List of students that registered for Spring 2015 but have not yet registered for Fall 2016.

   (Area of Study Title, StudentID, LastName, StudentID)

[4] Chapter 8 Order by, Logical operators
[5] Chapter 5 (Update) and Chapter 8 logical operator
[6] Chapter 9 Joining multiple tables
[7] Chapter 11
[8] Chapter 9 and more…

9. [Degree Audit Completed Report] List the students' completed courses for their first Area of Study. Run this query twice: Foss Winterlourn – 57, then Morganne Maynard – 200

   (StudentID, LastName, FirstName, AOSID, AOSIDTitle, DeptID, CourseNumber, FinalGrade)

   make sure you know what you're looking for before you start looking!

10. [Letter Grade GPA Conversion][9] Write a query to read the Enrollment table and convert the Letter Final Grade to its appropriate GPA numeric value.

   (EnrollmentID, StudentID, ClassSectID, FinalGrade, GPA) Sort by EnrollmentID.

Letter Grade	Grade Point	Percentage
A	4.0	95 – 100%
A-	3.7	90 – 94%
B+	3.3	87 – 89%
B	3.0	83 – 86%
B-	2.7	80 – 82%
C+	2.3	77 – 79%
C	2.0	73 – 76%
C-	1.7	70 – 72%
D+	1.3	67 – 69%
D	1.0	63 – 66%
D-	0.7	60 – 62%
F	0.0	0 -59%

11. [GPA for Each Student][10] Use #10 as your starting point. Create a query that list of students with their GPA (that has taking at least one class)

   (LastName, FirstName, StudentID, GPA

[9] Chapter 10 the CASE Expression See figure 10-41 p. 387
[10] At least Chapter 9(join) and 11(group)…

12. Create the required SQL query to extract the required data to display the Demographic form for an individual student. Run and document this query using student IDs: 469, 479, 532, & 964. Hint: You will need to run this query four times. Also note that we do have the data that is strike through on this form.

Student Biographical Information

Desilets, Wendy - ID: 123456

Profile

ID Number:	123456		
Soc Sec Number:	XXX-XX-0088[11]	Birthdate:	09/02/1990
Gender:	F	~~Ethnic:~~	~~Other~~
Address:	102 Spring Rd. Worcester, Ma, 01603	Phone:	508-555-5467

Academic

Advisor:	Desilets, Robert	Acad Status:	Good standing
Degree:	Certificate	Classification:	Sophomore[12]
Major1:	Web Applications Cert	Major2:	CIS Career Enterprise
Catalog:	Undergraduate 2020-2021	~~Max Hours:~~	~~19.00~~

13. You must also create an Operational Academic Transaction System Reference Manual[13] which details all the technical details about your SQL code and database. "How to and Why" Make sure to include the detailed outputs from your initial load steps (Create, Insert, & Alter) as separate Appendixes.

The following sample shows you the proper output format. I left the table borders in place to show you the alignment. You should hide the borders.

[11] Extract the last four digits of the SSN using the SUBSTR function

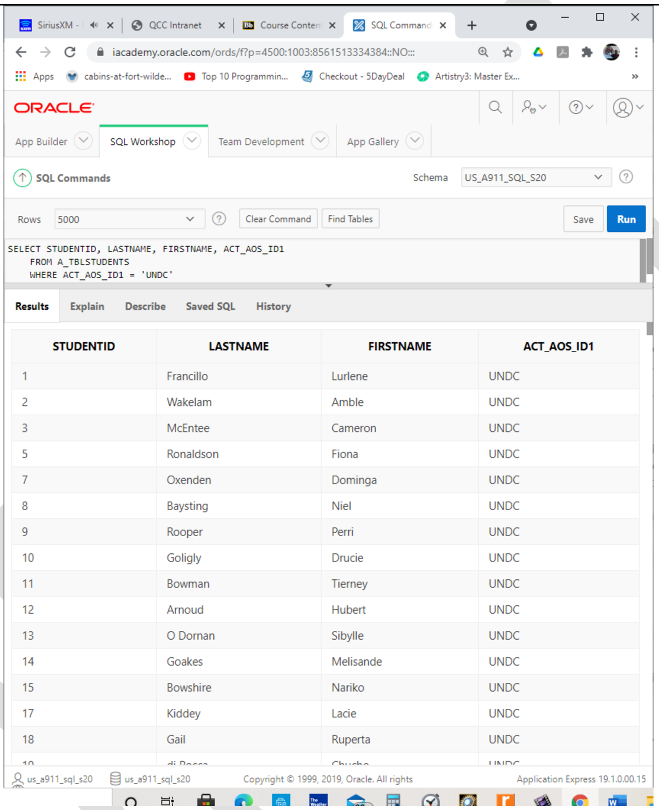
[12] Calculate based on < 30 credits Classification = Freshman, >= 30 = ~~freshman~~ sophmore

[13] This document must use formal standards. It MUST include a formal Title page, TOC, and the appropriate Appendixes

Sample Output Format

#A-99 [List all undeclared students][14]

```
SELECT STUDENTID, LASTNAME, FIRSTNAME, ACT_AOS_ID1
    FROM A_TBLSTUDENTS
    WHERE ACT_AOS_ID1 = 'UNDC'
```



Schema US_A911_SQL_S20

Rows 5000

```
SELECT STUDENTID, LASTNAME, FIRSTNAME, ACT_AOS_ID1
    FROM A_TBLSTUDENTS
    WHERE ACT_AOS_ID1 = 'UNDC'
```

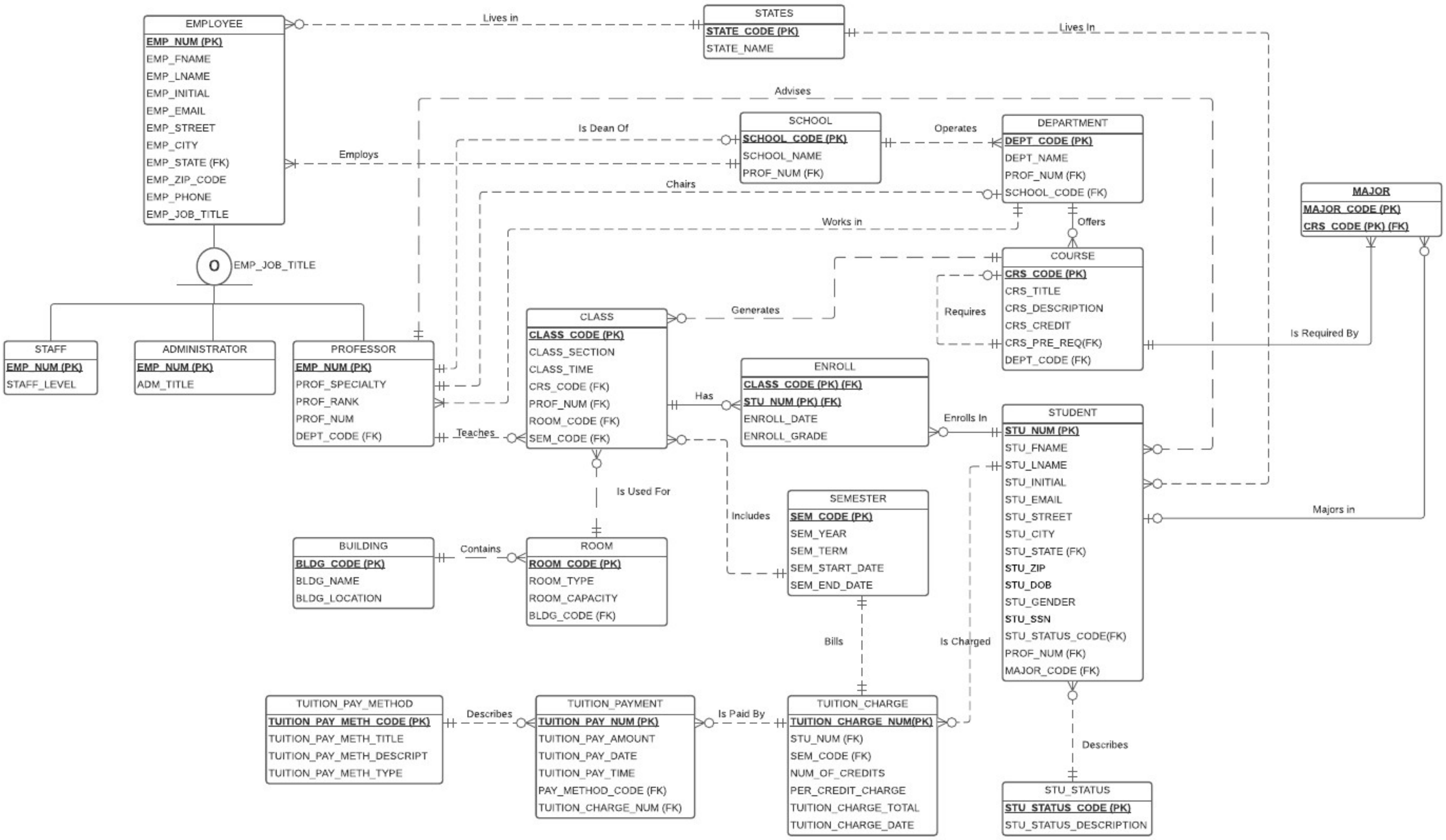
STUDENTID	LASTNAME	FIRSTNAME	ACT_AOS_ID1
1	Francillo	Lurlene	UNDC
2	Wakelam	Amble	UNDC
3	McEntee	Cameron	UNDC
5	Ronaldson	Fiona	UNDC
7	Oxenden	Dominga	UNDC
8	Baysting	Niel	UNDC
9	Rooper	Perri	UNDC
10	Goligly	Drucie	UNDC
11	Bowman	Tierney	UNDC
12	Arnoud	Hubert	UNDC
13	O Dornan	Sibylle	UNDC
14	Goakes	Melisande	UNDC
15	Bowshire	Nariko	UNDC
17	Kiddey	Lacie	UNDC
18	Gail	Ruperta	UNDC
19	di Rocca	Chucho	UNDC

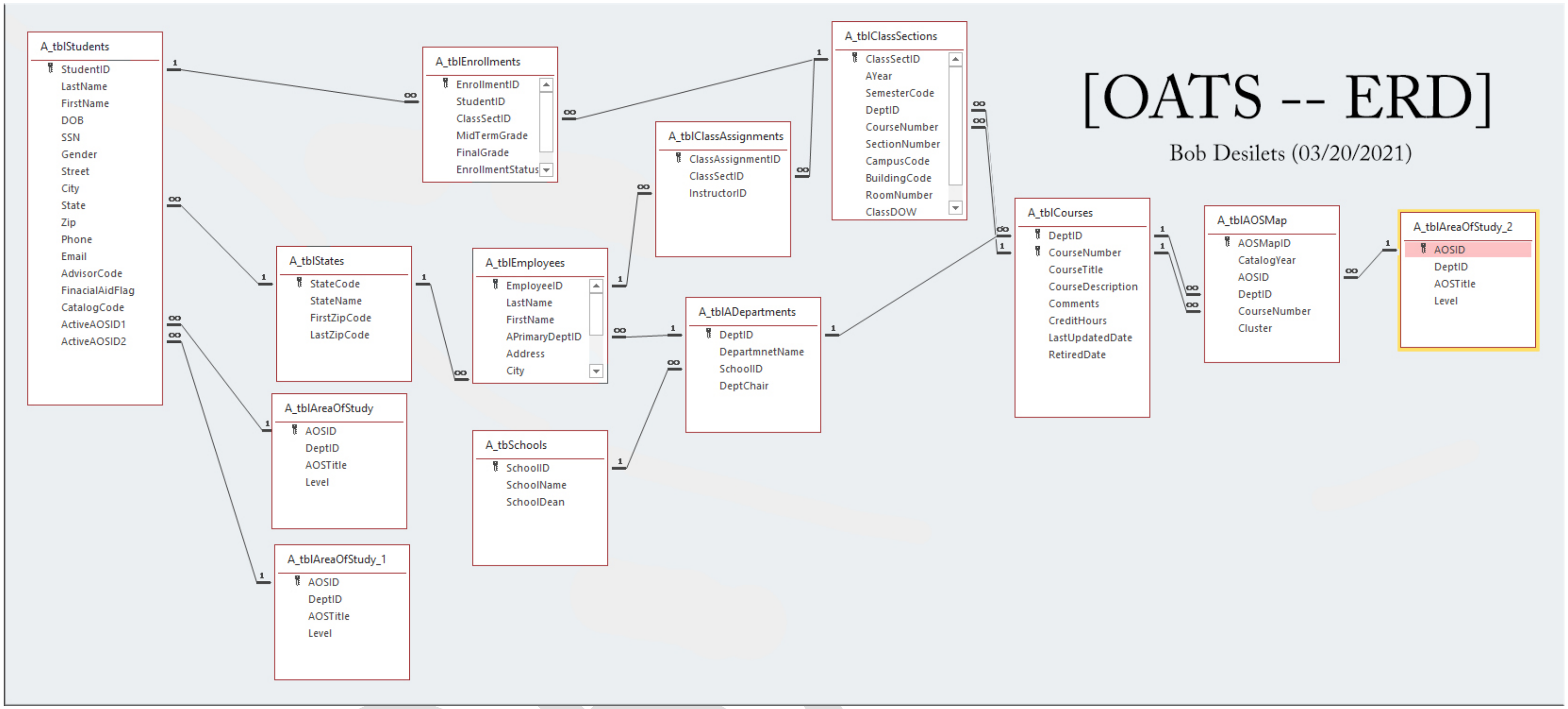
us_a911_sql_s20 us_a911_sql_s20 Copyright © 1999, 2019, Oracle. All rights Application Express 19.1.0.00.15

[14] #A-99 [List all undeclared students] This is an example of how to document each problem using footnotes

Entity Relational Diagram [ERD] Draft

Entity Relational Diagram with Business Rules





[OATS -- ERD]

Bob Desilets (03/20/2021)

Appendix

Appendix A: SQL Commands & Statements Command Covers in the Class

The follow list tells you to which SQL statements and functionality that we have covered in this class. Try to incorporate as many as you can into your project. Unless otherwise request you might find yourself not using all the items on this list. Remember to label the statement/functions in your manual to match the list below for cross reference purposes.[15] You are not required to do all these, but please try to fit in as many as you can. You must do all of section B.

2. CREATE TABLE
3. CREATE CONSTRAINTS
    A. PRIMARY KEY
    B. FOREIGN KEY
    C. UNIQUE
    D. CHECK
    E. NOT NULL
    F. Display all Constraint in the Appendix-A
4. INSERT
5. Handling Virtual Columns
6. Modifying Existing Rows A. UPDATE command
    B. Substitution Variables
7. Deleting rows
8. Sequences
    A. Creating Sequences
    B. Altering Sequence Definitions
    C. Display all Sequences in the Appendix-B
9. Indexes
    A. B-Tree Indexes
    B. Bitmap Indexes
    C. Function-Based Indexes
    D. Display all Indexes in the Appendix-C

[15] You must use a footnote for each of the required functions from the list (Only footnote it the first time you make reference to it)

10. Creating Queries restricting Rows and Sorting Data
    - A. WHERE Clause
    - B. BETWEEN . . . AND Operator
    - C. IN Operator
    - D. LIKE Operator
    - E. Logical operators
    - F. Treatment of NULL Values
    - G. ORDER BY Clause
        - a) Secondary Sort
        - b) Sorting by SELECT Order

10. Joining Data from Multiple Tables
    - A. Cartesian Join
    - B. Equality Join
    - C. Non-Equality Join
    - D. Self-Join
    - E. Outer Join

11. Selected Single-Row Functions
    - A. Case Conversion [at least 1]
        - a) LOWER Function
        - b) UPPER FUNCTION
        - c) INITCAP Function
    - B. Character Manipulation Function [at least 3]
        - a) SUBSTR Function
        - b) INSTR Function
        - c) LENGTH Function
        - d) LPAD and RPAD Function
        - e) LTRIM and RTRIM Function
        - f) REPLACE Function
        - g) TRANSLATE Function
        - h) CONCAT Function
    - C. Number Functions [at least 2]
        - a) ROUND
        - b) TRUNCATE
        - c) MOD Function

d) ABS Function
e) POWER Function

D. DATE Function [at least 3]

a) MONTH_BETWEEN Function
b) ADD_MONTH Function
c) NEXT_DAY and LAST_DAY Function
d) TO_DATE Function
e) Rounding Date Values
f) Truncating Date Values
g) CURRENT_DATE Versus SYSDATE

12. GROUP Function [at least 2]

A. SUM Function
B. AVG Function
C. COUNT Function
D. MAX Function
E. MIN Function

13. Sub Queries and MERGE

A. Single-Row Sub query

Multiple-Row Sub Query

Appendix B: Sample Database Script

[Note: this is a sample… Make sure to use the latest script file provided you by your instructor]

```
DROP TABLE B_tblStates CASCADE CONSTRAINTS;

DROP TABLE B_tblSchools CASCADE CONSTRAINTS;

DROP TABLE B_tblDepartments CASCADE CONSTRAINTS;

DROP TABLE B_tblArea_Of_Studys CASCADE CONSTRAINTS;

DROP TABLE B_tblStudents CASCADE CONSTRAINTS;

DROP TABLE B_tblEmployees CASCADE CONSTRAINTS;

DROP TABLE B_tblCourses CASCADE CONSTRAINTS;

DROP TABLE B_tblClass_Assignments CASCADE CONSTRAINTS;

DROP TABLE B_tblAOS_Maps CASCADE CONSTRAINTS;

DROP TABLE B_tblClass_Sections CASCADE CONSTRAINTS;

DROP TABLE B_tblEnrollments CASCADE CONSTRAINTS;



CREATE TABLE B_tblStates

   (   STATE_CODE         CHAR(2),

       STATE_NAME         VARCHAR2(25),

       FIRST_ZIP_CODE      VARCHAR2(10),

       LAST_ZIP_CODE       VARCHAR2(10),

           CONSTRAINT STATES_STATESCODE_PK PRIMARY KEY
(STATE_CODE)

   );



CREATE TABLE B_tblAOS_Maps

   (   AOS_MAPID          NUMBER (6),
    CATALOG_YEAR       NUMBER (4)       NOT NULL,
    AOSID              CHAR(4)          NOT NULL,
    DEPTID             CHAR(3)          NOT NULL,
    COURSE_NUMBER      CHAR(3)          NOT NULL,
    SEMESTER_CLUSTER   NUMBER(2)        DEFAULT 0,
      CONSTRAINT AOSMAPS_MAPID_PK PRIMARY KEY (AOS_MAPID)
  );

CREATE TABLE B_tblSchools
  (  SCHOOLID          VARCHAR2(10),
    SCHOOL_NAME        VARCHAR2(200) NOT NULL,
    DEANID             NUMBER(6,0),
      CONSTRAINT SCHOOLS_SCHOOLID_PK PRIMARY KEY (SCHOOLID)
  );

CREATE TABLE B_tblDepartments
  (  DEPTID            CHAR(3),
    DEPARTMENT_NAME    VARCHAR2(200) NOT NULL,
    SCHOOLID           VARCHAR2(10),
    DEPT_CHAIRID       NUMBER(6,0),
      CONSTRAINT DEPARTMENTS_DEPTID_PK PRIMARY KEY (DEPTID)


  );

CREATE TABLE B_tblArea_Of_Studys
  (  AOSID             CHAR(4),
    DEPTID             CHAR(3)          NOT NULL,
     AOSTITLE        VARCHAR(200)    NOT NULL,
     AOS_LEVEL        CHAR(1),
        CONSTRAINT AOS_AOSID_PK PRIMARY KEY (AOSID)
  );



CREATE TABLE B_tblStudents
  (    STUDENTID          NUMBER(6,0),
       LAST_NAME           VARCHAR2(25)   NOT NULL,
       FIRST_NAME          VARCHAR2(25)   NOT NULL,
       DOB                DATE           NOT NULL,
       SSN                VARCHAR2(11)   NOT NULL,
       GENDER               VARCHAR2(1),
       STREET              VARCHAR2(100)  NOT NULL,
       CITY               VARCHAR2(50)   NOT NULL,
       STATE_CODE          CHAR(2),
       ZIP_CODE            VARCHAR2(10),
       PHONE               VARCHAR2(12),
       EMAIL              VARCHAR2(200),
       ADVISOR_CODE       NUMBER(6,0),
       FINACIAL_AID_FLAG  VARCHAR2(5)    DEFAULT 'FALSE',
       CATALOG_CODE       VARCHAR2(50),
       ACT_AOS_ID1        CHAR(4)        DEFAULT 'UNDC',
       ACT_AOS_ID2        CHAR(4),
          CONSTRAINT STUDENT_STUDENID_PK PRIMARY KEY
(STUDENTID),
          CONSTRAINT STUDENT_SSN_UK UNIQUE (SSN)
   );



CREATE TABLE B_tblEmployees
   (   EMPLOYEEID          NUMBER(6,0),
       LAST_NAME           VARCHAR2(25)    NOT NULL,
       FIRST_NAME          VARCHAR2(25)    NOT NULL,
       A_PRIMARY_ADEPTID   CHAR(3),
       ADDRESS             VARCHAR2(100),
       CITY                VARCHAR2(50),
       STATE_CODE          CHAR(2),
       ZIP_CODE            VARCHAR2(10),
       PHONE               VARCHAR2(12),
       EMAIL               VARCHAR2(200),
          CONSTRAINT EMPLOYEES_EMPLOYEEID_PK PRIMARY KEY
(EMPLOYEEID)
   );



CREATE TABLE B_tblCourses
   (   DEPTID              CHAR(3),
       COURSE_NUMBER       CHAR(3),
       COURSE_TITLE        VARCHAR2(200)     NOT NULL,
       COURSE_DESCRIPTION VARCHAR2(4000)    NOT NULL,
      COMMENTS          VARCHAR2(4000),
      CREDIT_HOURS       NUMBER(3),
      LAST_UPDATED_DATE  DATE,
      RETIRED_DATE       DATE,
         CONSTRAINT DEPTID_COURSE_NO_PK PRIMARY KEY (DEPTID,
COURSE_NUMBER)
   );

CREATE TABLE B_tblClass_Assignments
   (   CLASS_ASSIGNMENTID NUMBER(6),
      CLASS_SECTID       NUMBER(6)       NOT NULL,
      INSTRUCTORID       NUMBER(6)       NOT NULL,
         CONSTRAINT CASSIGNMENTS_ASSIGNID_PK PRIMARY KEY
(CLASS_ASSIGNMENTID)
   );

CREATE TABLE B_tblClass_Sections
   (   CLASS_SECTID       NUMBER(6),
      B_YEAR             NUMBER(4)       NOT NULL,
      SEMESTER_CODE      NUMBER(2)       NOT NULL,
      DEPTID             CHAR(3)         NOT NULL,
      COURSE_NUMBER      CHAR(3)         NOT NULL,
      SECTION_NUMBER     CHAR(2)         NOT NULL,
      CAMPUS_CODE        CHAR(5),
      BUILDING_CODE      VARCHAR2(10),
      ROOM_NUMBER        VARCHAR2(10),
      CLASS_DOW          VARCHAR2(7),
      CLASS_TIME         VARCHAR2(13),
        CONSTRAINT CLASSSECT_SECTID_PK PRIMARY KEY
(CLASS_SECTID)
   );


CREATE TABLE B_tblEnrollments
   (  ENROLLMENTID     NUMBER(6),
      STUDENTID        NUMBER(6)       NOT NULL,
      CLASS_SECTID     NUMBER(6)       NOT NULL,
      MID_TERM_GRADE   CHAR(2),
      FINAL_GRADE      CHAR(2),
      ENROLLMENT_STATUS VARCHAR2(10)    DEFAULT 'GRADED',
      REGISTRATION_DATE DATE,
        CONSTRAINT ENROLLMENTS_ENROLLMENTID_PK PRIMARY KEY
(ENROLLMENTID)
   );
```

[INSERT]

See the provided script for the actual data INSERT statements

```
ALTER TABLE B_tblDepartments
   DROP CONSTRAINT DEPARTMENT_SCHOOLID_FK;


ALTER TABLE B_tblDepartments
   DROP CONSTRAINT DEPARTMENT_DEPT_CHAIRID_FK;


ALTER TABLE B_tblArea_Of_Studys
   DROP CONSTRAINT MAJORS_DEPTID_FK;


ALTER TABLE B_tblStudents
```

```
   DROP CONSTRAINT STUDENTS_STATECODE_FK;

ALTER TABLE B_tblStudents
   DROP CONSTRAINT STUDENTS_AAOS_1_FK;

ALTER TABLE B_tblStudents
   DROP CONSTRAINT STUDENTS_AAOS_2_FK;

ALTER TABLE B_tblEmployees
   DROP CONSTRAINT EMPLOYEES_STATECODE_FK;

ALTER TABLE B_tblEmployees
   DROP CONSTRAINT EMPLOY_A_PRIMARY_DEPTID_FK;

ALTER TABLE B_tblCourses
   DROP CONSTRAINT COURSES_DEPTID_FK;

ALTER TABLE B_tblClass_Assignments
   DROP CONSTRAINT CASSIGNMENTS_CSECT_FK;

ALTER TABLE B_tblClass_Assignments
   DROP CONSTRAINT CASSIGNMENTS_INSTRUCTORID_FK;

ALTER TABLE B_tblClass_Sections
   DROP CONSTRAINT CLASSSECT_SEMESTERCODE_FK;

ALTER TABLE B_tblEnrollments
```

```
   DROP CONSTRAINT ENROLLMENTS_CLASSSECTID;


ALTER TABLE B_tblDepartments
   ADD CONSTRAINT DEPARTMENT_SCHOOLID_FK  FOREIGN KEY (SCHOOLID)
          REFERENCES B_tblSchools (SCHOOLID);


ALTER TABLE B_tblDepartments
   ADD CONSTRAINT DEPARTMENT_DEPT_CHAIRID_FK  FOREIGN KEY (DEPT_CHAIRID)
          REFERENCES B_tblEmployees (EMPLOYEEID);


ALTER TABLE B_tblArea_Of_Studys
   ADD    CONSTRAINT MAJORS_DEPTID_FK  FOREIGN KEY (DEPTID)
            REFERENCES B_tblDepartments (DEPTID);


ALTER TABLE B_tblStudents
   ADD    CONSTRAINT STUDENTS_STATECODE_FK  FOREIGN KEY (STATE_CODE)
            REFERENCES B_tblStates (STATE_CODE);


ALTER TABLE B_tblStudents
   ADD    CONSTRAINT STUDENTS_AAOS_1_FK  FOREIGN KEY (ACT_AOS_ID1)
            REFERENCES B_tblArea_Of_Studys (AOSID);


ALTER TABLE B_tblStudents
   ADD    CONSTRAINT STUDENTS_AAOS_2_FK  FOREIGN KEY (ACT_AOS_ID2)
```

```
            REFERENCES B_tblArea_Of_Studys (AOSID);


ALTER TABLE B_tblEmployees
   ADD    CONSTRAINT EMPLOYEES_STATECODE_FK  FOREIGN KEY
(STATE_CODE)
            REFERENCES B_tblStates (STATE_CODE);


ALTER TABLE B_tblEmployees
   ADD    CONSTRAINT EMPLOY_B_PRIMARY_DEPTID_FK  FOREIGN KEY
(A_PRIMARY_ADEPTID)
            REFERENCES B_tblDepartments (DEPTID);


ALTER TABLE B_tblCourses
   ADD       CONSTRAINT COURSES_DEPTID_FK  FOREIGN KEY (DEPTID)
            REFERENCES B_tblDepartments (DEPTID);


ALTER TABLE B_tblClass_Assignments
   ADD   CONSTRAINT CASSIGNMENTS_CSECT_FK  FOREIGN KEY
(CLASS_SECTID)
            REFERENCES B_tblClass_Sections (CLASS_SECTID);


ALTER TABLE B_tblClass_Assignments
   ADD       CONSTRAINT CASSIGNMENTS_INSTRUCTORID_FK  FOREIGN
KEY (INSTRUCTORID)
            REFERENCES B_tblEmployees (EMPLOYEEID);


ALTER TABLE B_tblClass_Sections
   ADD       CONSTRAINT CLASSSECT_SEMESTERCODE_FK FOREIGN KEY
(DEPTID)
```

```
            REFERENCES B_tblDepartments (DEPTID);


ALTER TABLE B_tblEnrollments
   ADD       CONSTRAINT ENROLLMENTS_CLASSSECTID FOREIGN KEY
(CLASS_SECTID)
            REFERENCES B_tblClass_Sections (CLASS_SECTID);
```