COMP3311 23T3 Database Systems

Assignment 2
The MyMyUNSW Database (SQL)

Last updated: **Monday 30th October 4:28pm**

Most recent changes are shown in red ... older changes are shown in brown.

[Specification] [Database] **[SQL Schema]** [Grades+Rules] [Examples] [Testing] [Submitting] [Fixes+Updates]

Introduction

This document contains an SQL schema for the MyMyUNSW database based on the ER model on the Database page.

Schema

Downloadable Version schema.sql

```
 1. -- COMP3311 23T3 Assignment 2
 2. -- Schema for MyMyUNSW database
 3.
 4. CREATE DOMAIN TermCode AS CHAR(4) CHECK (VALUE ~'[12][01
 5. CREATE DOMAIN SubjectCode AS CHAR(8) CHECK (VALUE ~ '[A-
 6. CREATE DOMAIN StreamCode AS CHAR(6) CHECK (VALUE ~ '[A-Z
 7. CREATE DOMAIN ProgramCode AS CHAR(4) CHECK (VALUE ~ '[0-
 8. CREATE DOMAIN ZidValue AS INTEGER CHECK (VALUE BETWEEN 1
 9. CREATE TYPE ResidentType AS enum ('AUS','AUSPR','INTL','
10. CREATE TYPE UnitType AS enum ('UNSW','faculty','school',
11. CREATE TYPE CareerType AS enum ('UGRD','PGRD','RSCH','NA
12. CREATE TYPE GradeType AS enum
13.         ('A', 'A+', 'A-', 'AF', 'AS', 'AW', 'B', 'B+', '
14.          'C-', 'CR', 'D', 'D+', 'D-', 'DN', 'E', 'E+', '
15.          'EM', 'F', 'FL', 'HD', 'NA', 'NC', 'NF', 'PE',
16.          'RC', 'RD', 'RS', 'SY', 'UF', 'WD', 'WJ', 'XE')
17. CREATE TYPE ReqType AS enum
18.         ('core','elective','free','gened','stream','uoc'
19.
20. CREATE TABLE Terms (
21.         id          INTEGER,
22.         code        TermCode NOT NULL,
23.         starting    DATE,
24.         ending      DATE,
25.         description text,
26.         PRIMARY KEY (id)
27. );
28.
29. CREATE TABLE Countries (
30.         id          INTEGER,
31.         code        CHAR(3) NOT NULL,
32.         name        text NOT NULL,
33.         PRIMARY KEY (id)
34. );
35.
36. CREATE TABLE People (
37.         id          INTEGER,
38.         zid         ZidValue UNIQUE NOT NULL,
39.         family_name text,
```

```
40.         given_names text,
41.         full_name   text,
42.         origin      INTEGER REFERENCES Countries(id),
43.         PRIMARY KEY (id)
44. );
45.
46. CREATE TABLE Students (
47.         id          INTEGER REFERENCES People(id),
48.         STATUS      ResidentType,
49.         PRIMARY KEY (id)
50. );
51.
52. CREATE TABLE Staff (
53.         id          INTEGER REFERENCES People(id),
54.         PRIMARY KEY (id)
55. );
56.
57. CREATE TABLE Orgunits (
58.         id          INTEGER,
59.         code        VARCHAR(10) NOT NULL,
60.         name        text NOT NULL,
61.         utype       UnitType NOT NULL,
62.         parent      INTEGER REFERENCES Orgunits(id),
63.         PRIMARY KEY (id)
64. );
65.
66. CREATE TABLE Subjects (
67.         id          INTEGER,
68.         code        SubjectCode NOT NULL,
69.         title       text NOT NULL,
70.         uoc         INTEGER NOT NULL CHECK (uoc BETWEEN (
71.         career      CareerType,
72.         owner       INTEGER NOT NULL REFERENCES Orgunits
73.         PRIMARY KEY (id)
74. );
75.
76. CREATE TABLE Courses (
77.         id          INTEGER,
78.         subject     INTEGER NOT NULL REFERENCES Subjects
79.         term        INTEGER NOT NULL REFERENCES Terms(id
80.         convenor    INTEGER REFERENCES Staff(id),
81.         satisfact   INTEGER CHECK (satisfact BETWEEN 0 AI
82.         nresponses  INTEGER,
83.         PRIMARY KEY (id)
84. );
85.
86. CREATE TABLE Streams (
87.         id          INTEGER,
88.         code        StreamCode,
89.         name        text, -- not null,
90.         PRIMARY KEY (id)
91. );
92.
93. CREATE TABLE Programs (
94.         id          INTEGER,
95.         code        ProgramCode,
96.         name        text, -- not null,
97.         PRIMARY KEY (id)
98. );
99.
```

```
100. CREATE TABLE Program_enrolments (
101.         id          INTEGER,
102.         student     INTEGER REFERENCES Students(id),
103.         term        INTEGER REFERENCES Terms(id),
104.         program     INTEGER REFERENCES Programs(id),
105.         PRIMARY KEY (id)
106. );
107.
108. CREATE TABLE Stream_enrolments (
109.         part_of     INTEGER REFERENCES Program_enrolment
110.         stream      INTEGER REFERENCES Streams(id),
111.         PRIMARY KEY (part_of,stream)
112. );
113.
114. CREATE TABLE Course_enrolments (
115.         student     INTEGER REFERENCES Students(id),
116.         course      INTEGER REFERENCES Courses(id),
117.         mark        INTEGER CHECK (mark BETWEEN 0 AND 10
118.         grade       GradeType,
119.         PRIMARY KEY (student,course)
120. );
121.
122. CREATE TABLE Requirements (
123.         id          INTEGER,
124.         name        text NOT NULL,
125.         rtype       ReqType NOT NULL,
126.         min_req     INTEGER CHECK (min_req > 0),
127.         max_req     INTEGER CHECK (max_req > 0),
128.         acadobjs    text,
129.         for_stream  INTEGER REFERENCES Streams(id),
130.         for_program INTEGER REFERENCES Programs(id),
131.         CONSTRAINT  OneOrOther CHECK (
132.                                 for_stream IS NULL AND fo
133.                                 OR
134.                                 for_stream IS NOT NULL AN
135.                                 ),
136.         PRIMARY KEY (id)
137. );
```