```
1
2
 Zacharie Happel
3
 Database Management Final Project
5
6
  7
8
  9
  10
  11
12
13
  /* Each person is given a unique CampusWideID,
14
 */
15
16
17
18
  USE College DB;
19
  CREATE TABLE Person (
20
   CampusWideID INT NOT NULL UNIQUE AUTO INCREMENT,
21
   FirstName VARCHAR(20) DEFAULT NULL,
22
   LastName VARCHAR(20) DEFAULT NULL,
23
   MiddleName VARCHAR(20) DEFAULT NULL,
24
   MaidenName VARCHAR(20) DEFAULT NULL,
25
   SocialSecurity INT DEFAULT NULL,
26
   DoB VARCHAR(8) NOT NULL,
27
   BannerID VARCHAR(20) DEFAULT NULL,
28
   Gender VARCHAR(20) DEFAULT NULL,
29
   Sex VARCHAR(10) DEFAULT NULL,
30
   Race VARCHAR(25) DEFAULT NULL,
31
   Role VARCHAR(20) NOT NULL ,
32
   HomeAddress VARCHAR(70) DEFAULT NULL,
33
   SecondaryAddress VARCHAR(70) DEFAULT NULL,
34
   PrimaryPhone VARCHAR(14) DEFAULT NULL,
35
   SecondaryPhone VARCHAR(14) DEFAULT NULL,
36
37
38
   PRIMARY KEY (CampusWideID)
39
40
41
42
  43
  \/\/\Population/\/\/\//
44
```

```
46
47
48
49
50
51
52
53
54
55
56
57
                          Insertion of people into Person table
58
                 (Redacted because it contained personal information)
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
  80
81
  \/\/\/\Update/\/\/\/\//
82
  83
84
  /* Enters the BannerID's for the first time */
85
  UPDATE Person
86
  SET BannerID = (SELECT Concat( LOWER(SUBSTRING(Title, 1, 2)) , 00,
87
88 UPPER(SUBSTRING(FirstName, 1, 2)), UPPER(SUBSTRING(LastName, 1, 2)),
  00000, CampusWideID) as Banner
  WHERE BannerID IS NULL )
```

```
WHERE CampusWideID IS NOT NULL;
92
93
94 /* Can be used if BannerID format needs to be changed */
95 UPDATE Person
  |SET BannerID = (SELECT Concat( LOWER(SUBSTRING(Title,1, 2)) , 00,
97 UPPER(SUBSTRING(FirstName, 1, 2)), UPPER(SUBSTRING(LastName, 1, 2)),
98 00000, CampusWideID) as Banner
  WHERE BannerID IS NOT NULL)
100 WHERE CampusWideID IS NOT NULL;
101
102
103 /* Solves the isse of some BannerID's being null while others are not*/
104 UPDATE Person
105 | SET BannerID = (SELECT Concat( LOWER(SUBSTRING(Title, 1, 2)) , 00,
106 UPPER(SUBSTRING(FirstName, 1, 2)), UPPER(SUBSTRING(LastName, 1, 2)),
107 00000, CampusWideID) as Banner
108 WHERE BannerID IS NULL OR BannerID IS NOT NULL;
  WHERE CampusWideID IS NOT NULL;
109
110
111
116
117
118
119
120
121 CREATE TABLE Semester(
122 SemesterID INT NOT NULL UNIQUE,
123 BeginDate VARCHAR (5),
124 EndDate VARCHAR (5),
125 Season VARCHAR(6),
126 PRIMARY KEY (SemesterID)
127 | );
128
129
132
133
134 INSERT INTO Semester
  VALUES (10 , '12/20', '01/23', 'Winter');
135
```

```
136
137
  INSERT INTO Semester
  VALUES (20 , '01/25', '5/25', 'Spring');
138
139
  INSERT INTO Semester
140
  VALUES (30 , '06/05', '08/23', 'Summer');
141
142
143 INSERT INTO Semester
  VALUES (40 , '8/28', '12/16', 'Fall');
145
146
147
  148
149
  \/\/\/\/\/\/\/\/\/\/\/\/\/\/\/
  150
  /\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/
  152
153
154
155
156
157 CREATE TABLE Department(
158 DepartmentID INT NOT NULL AUTO INCREMENT,
159 DepartmentName VARCHAR(35),
160
  PRIMARY KEY (DepartmentID)
161
  );
162
163
164
165
166
  \/\/\Population/\/\/\//
167
  168
169
  INSERT INTO Department (DepartmentName)
170
  VALUES ('Arts Department'),
171
               ('Computer Science Department'),
172
               ('Fashion Department'),
173
               ('Medical Department');
174
175
176
177
178
179
180
```

```
181
182
183
  184
  185
186
  187
188
189
190
191
192 CREATE TABLE Course (
193 CourseID INT NOT NULL UNIQUE AUTO INCREMENT,
194 DepartmentID INT,
195 | IlearnID VARCHAR(30) UNIQUE DEFAULT NULL,
196 CourseName VARCHAR(50) NOT NULL,
197 CreditAmt INT NOT NULL,
198 SemesterID INT,
199
  PRIMARY KEY (CourseID),
200
  FOREIGN KEY (SemesterID) REFERENCES Semester(SemesterID),
202 FOREIGN KEY (DepartmentID) REFERENCES Department(DepartmentID)
  );
203
204
205
206
  \/\/\Population/\/\/\/\/
207
  208
209
  INSERT INTO Course(DepartmentID, CourseName, CreditAmt)
  VALUES (2, 'Intro to Programming', 4),
211
                (2, 'Software Development I', 4),
212
                (2, 'Software Development II', 4),
213
                ( 2, 'Internetworking', 4),
214
                (2, 'Database Management', 4),
215
                (2, 'Data Communications', 4);
216
217
  INSERT INTO Course(DepartmentID, CourseName, CreditAmt)
218
  VALUES (3, 'Fashion Economics', 4),
219
                (3, 'Intro Design Studio', 4),
220
                (4, 'Pathogens and Microbes', 4),
221
                (4, 'Fundamentals of Immunology', 4),
222
                (4, 'Radiology Technology', 4),
223
                (1, 'Watching Paint Dry', 4);
224
225
```

```
226
227
228
229
230
232 \/\/\Complex Queries ////
233 /\/\/\/\/\/\/\/\/\/\/\/\/
234
   /* These queries allocate an individual, and unique IlearnID to each course
235
       using the unDepartment ID number,
236
237
238 UPDATE Course
239 | SET IlearnID = (SELECT Concat(DepartmentID, UPPER(SUBSTRING(CourseName, 1, 5)
240 AS Ilearn
241 WHERE CourseName = 'Software Development I')
242 WHERE CourseName = 'Software Development I';
243
244
245 UPDATE Course
246 | SET IlearnID = (SELECT Concat(DepartmentID, UPPER(SUBSTRING(CourseName, 1, 5)
247 AS Ilearn
248 WHERE CourseName = 'Software Development I')
249 WHERE CourseName = 'Software Development I';
250
251
252 UPDATE Course
253 SET IlearnID = (SELECT Concat(DepartmentID, UPPER(SUBSTRING(CourseName, 1, 5)
254 as Ilearn
255 WHERE CourseName = 'Intro to Programming')
256 WHERE CourseName = 'Intro to Programming';
257
258 UPDATE Course
259 SET IlearnID = (SELECT Concat(DepartmentID, UPPER(SUBSTRING(CourseName, 1, 5)
260 as Ilearn
261 WHERE CourseName = 'Internetworking')
262 WHERE CourseName = 'Internetworking';
263
264 UPDATE Course
265 SET IlearnID = (SELECT Concat(DepartmentID, UPPER(SUBSTRING(CourseName, 1, 5)
266 as Ilearn
267 WHERE CourseName = 'Database Management')
268 WHERE CourseName = 'Database Management';
269
270 UPDATE Course
```

```
271 | SET IlearnID = (SELECT Concat(DepartmentID, UPPER(SUBSTRING(CourseName, 1, 5)
272 as Ilearn
273 WHERE CourseName = 'Data Communications')
274 WHERE CourseName = 'Data Communications';
275
276
277
278
279
280
  281
  \/\/\/\/\/\/\/\/\/\/\/\/
  282
283
  284
285
286
287
288
  CREATE TABLE CourseInstance(
289
290 IlearnID VARCHAR(30) UNIQUE NOT NULL,
  SemesterID INT DEFAULT NULL,
292 InstructorCWID INT DEFAULT NULL,
293
294
295 FOREIGN KEY (IlearnID) REFERENCES Course(IlearnID),
  FOREIGN KEY (SemesterID) REFERENCES Semester(SemesterID)
296
  );
297
298
299
300
301
  302
  \/\/\/\/INSERT\/\/\/\/\/
  303
304
305
306
  INSERT INTO CourseInstance(IlearnID, SemesterID)
  VALUES ('1WATCH12', 20),
308
309
               ('2DATA 6', 20),
               ('2DATAB5', 20),
310
311
               ('2INTER4', 20),
               ('2INTR01', 20),
312
313
               ('2S0FTW2', 20),
               ('2SOFTW3', 20),
314
               ('3FASHI7',20),
315
```

```
316
                    ('3INTR08',20),
                    ('4FUNDA10',20),
317
                    ('4PATH09',20),
318
                    ('4RADI011',20),
319
                    ('1WATCH12', 40),
320
                    ('2DATA 6', 40),
321
                    ('2DATAB5', 40),
322
                    ('2INTER4', 40),
323
                    ('2INTR01', 40),
324
                    ('2SOFTW2', 40),
325
                    ('2SOFTW3', 40),
326
                    ('3FASHI7',40),
327
                    ('3INTRO8',40),
328
                    ('4FUNDA10',40),
329
                    ('4PATH09',40),
330
                    ('4RADI011',40);
331
        /* Copies these IlearnID's from the ones that were previously generated
332
333
334
335
336
   337
   \/\/\Complex Queries\/\/\
338
   339
340
341
342
   SELECT ci.IlearnID
343
344 FROM CourseInstance ci
345 LEFT JOIN Course course ON ci.IlearnID = course.IlearnID
346 WHERE course DepartmentID = 2
   /* Returns IlearnID of course in Comp Sci */
347
348
349
   SELECT i.DepartmentID, i.CampusWideID FROM Instructor i
350
   INNER JOIN Course c
351
352
       ON i.DepartmentID = c.DepartmentID
       INNER JOIN CourseInstance ci
353
           ON c.IlearnID = ci.IlearnID
354
           WHERE c.DepartmentID = 2;
355
356
   /* Returns the department ID associated with all Courses in CourseInstance,
357
      the Instructor that teaches for that department
358
359
360
```

```
361
362
363
  /*
364
  365
  366
  367
  368
  369
370
371
372
373
 CREATE TABLE StudentCourseRegistry (
374
 IlearnID VARCHAR(30) NOT NULL,
375
 BannerID VARCHAR(20) DEFAULT NULL,
376
377
 PRIMARY KEY (IlearnID),
378
 FOREIGN KEY (IlearnID) REFERENCES Course(IlearnID),
379
 FOREIGN KEY (BannerID) REFERENCES Person(BannerID)
380
  );
381
382
383
384
385
386
387
388
  389
  390
  391
392
  393
394
395
396
397
 CREATE TABLE Instructor (
398
 CampusWideID INT NOT NULL UNIQUE AUTO INCREMENT,
399
 BannerID VARCHAR(20) DEFAULT NULL,
400
 FirstName VARCHAR(20) DEFAULT NULL,
401
 LastName VARCHAR(20) DEFAULT NULL,
402
 DepartmentID INT(11),
403
404
 PRIMARY KEY (CampusWideID),
405
```

```
406 FOREIGN KEY (CampusWideID) REFERENCES Person(CampusWideID),
407 FOREIGN KEY (BannerID) REFERENCES Person(BannerID)
   FOREIGN KEY (DepartmentID) REFERENCES Department(DepartmentID)
408
   );
409
410
411
412
413
414
   415
   \/\/\/INSERT/Population/\//
416
   417
418
   UPDATE Instructor
419
   SET DepartmentID = 2
420
421 WHERE Instructor.LastName = 'Arias';
422
423 UPDATE Instructor
   SET DepartmentID = 1
424
425 WHERE Instructor.LastName = 'Valentine;
426
427 UPDATE Instructor
   SET DepartmentID = 3
428
   WHERE Instructor.LastName = 'Haley';
429
430
   UPDATE Instructor
431
   SET DepartmentID = 4
432
   WHERE Instructor.LastName = 'Sharpe';
433
434
435
436
437
438
   /\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/
439
   \/\/\/Complex Queries\/\/\/
440
   /\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/
441
442
   INSERT INTO Instructor(CampusWideID, BannerID, FirstName, LastName)
443
   SELECT p.CampusWideID, p.BannerID,
444
       p.FirstName,
445
       p.LastName
446
   FROM Person p
447
448
   LEFT JOIN Instructor s ON p.CampusWideID = s.CampusWideID
   WHERE p.Title = 'Instructor' AND s.CampusWideID IS NULL;
449
450
```

```
451
452
453
454
455
456
457
458
    /*
   459
460
   461
462
463
464
465
466
467
468
   CREATE TABLE Student (
469
   CampusWideID INT NOT NULL UNIQUE AUTO INCREMENT,
470
   BannerID VARCHAR(20) DEFAULT NULL,
471
472 FirstName VARCHAR(20) DEFAULT NULL,
473 | LastName VARCHAR(20) DEFAULT NULL,
474 AcademicYear CHAR(4),
   OnCampusStatus CHAR(1),
475
476
   PRIMARY KEY (CampusWideID),
477
   FOREIGN KEY (CampusWideID) REFERENCES Person(CampusWideID)
478
479
   );
480
481
   /*
482
483
   /\/\/\/\/\/\/\/\/\/\/\/\/\/\/
   \/\/\Population/\/\/\/\/
484
   485
486
487
   /* Adding column */
488
   ALTER TABLE Student
489
   ADD COLUMN DepartmenOfMajor INT(11) DEFAULT NULL AFTER OnCampusStatus;
490
491
492
   /* Add FK constraint to DepartmentOfMajor, references DepartmentID */
493
494 ALTER TABLE `College_DB`. `Student`
   ADD CONSTRAINT `FK_Student_DepartmentOfMajor` FOREIGN KEY (`DepartmentOfMajo
495
```

```
REFERENCES `College DB`.`Department`(`DepartmentID`);
497
498
499
   /* Manually assigned DepartmentID's as I didn't expect to have to use it ear
500
501 UPDATE `College_DB`.`Student` SET `DepartmentOfMajor` = '1' WHERE (`CampusWi
502 UPDATE `College_DB`.`Student` SET `DepartmentOfMajor` = '1' WHERE (`CampusWi
503 UPDATE `College DB`.`Student` SET `DepartmentOfMajor` = '2' WHERE (`CampusWi
504 UPDATE `College DB`.`Student` SET `DepartmentOfMajor` = '1' WHERE (`CampusWi
505
                             Did only this used the MySQL alter table GUI tool
   /* Etc. etc. etc.
506
                              for the manual insertion of data*/
507
508
509
510
   /* Updating Department/Major */
511
512
513 UPDATE Student
514 SET DepartmentOfMajor = 2
515 WHERE SUBSTRING(Student.FirstName, 1, 1) IN ('Z', 'A', 'B', 'C', 'D', 'E', 'F',
   /* 40 students */
516
517
518 UPDATE Student
519 SET DepartmentOfMajor = 4
520 WHERE SUBSTRING(Student.FirstName, 1, 1) IN ('J', 'L', 'K', 'M');
521 /* 27 students */
522
523 UPDATE Student
524 SET DepartmentOfMajor = 1
   WHERE SUBSTRING(Student.FirstName, 1, 1) IN ('L', 'N', '0', 'P');
525
                   /* 14 students */
526
527
528
   UPDATE Student
   SET DepartmentOfMajor = 3
529
   WHERE SUBSTRING(Student.FirstName, 1, 1) IN ('Q', 'R', 'S', 'T', 'U', 'V', 'W
530
           /* 14 students */
531
532
533
534
535
536
537
   538
539 \\\\\/Complex Queries\\\\\\
```

```
541
542
543
544
545
  INSERT INTO Student(CampusWideID)
546
  SELECT p.CampusWideID
547
548 FROM Person p
549 LEFT JOIN Student s ON p.CampusWideID = s.CampusWideID
  WHERE p.Title = 'Student' AND s.CampusWideID IS NULL;
550
551
   /* Populates Student with CWID's of those with the Role 'Student'
552
                0R
553
     Using the below, you can populate the Table with CampusWideID, BannerID,
554
                                                and LastName
555
556
   INSERT INTO Student(CampusWideID, BannerID, FirstName, LastName)
557
   SELECT p.CampusWideID, p.BannerID,
558
      p.FirstName,
559
      p.LastName
560
  FROM Person p
561
  LEFT JOIN Student s ON p.CampusWideID = s.CampusWideID
562
  WHERE p.Title = 'Student' AND s.CampusWideID IS NULL;
563
564
565
566
567
568
569
570
   /*
   571
      \/\/\/\/\/\/\/\/\/\/\/
572
573
   574
   575
576
577
578
  CREATE TABLE AcademicHistory(
579
  CampusWideID INT NOT NULL UNIQUE AUTO_INCREMENT,
580
  BannerID VARCHAR(50) UNIQUE DEFAULT NULL,
581
  TotalCredits INT,
582
583 TransferCredits INT,
584 GPA DECIMAL,
  HighSchoolGPA DOUBLE,
585
```

```
SATScore INT,
586
587
  ACTScore INT,
588
  PRIMARY KEY (CampusWideID),
589
  FOREIGN KEY (CampusWideID) REFERENCES Person(CampusWideID),
590
  FOREIGN KEY (BannerID) REFERENCES Person(BannerID)
591
  );
592
593
594
595
596
597
598
599
   /*
  600
  601
  |||||| Triggers that I attempted to make but was never successfully able to
602
  603
  604
605
606
607
608
609
  DELETE TRIGGER IF EXISTS TriggerAssignBannerID;
610
  CREATE TRIGGER TriggerAssignBannerID
611
  AFTER INSERT ON Person
612
  FOR EACH ROW
613
614
615
  BEGIN
616
617
618
  SET @cwid;
619
620
  SELECT CampusWideID INTO @cwid
621
  FROM Person
622
623 WHERE CampusWideID IS NOT NULL
  ORDER BY CampusWideID
624
  DESC LIMIT 0,1;
625
626
  SELECT @OptionalLocalDeclarationOfBannerID;
627
628
  INSERT INTO Testing(Numbers)
629
  value (@cwid);
630
```

```
631
   CALL AssignBannerID(@cwid, @OptionalLocalDeclarationOfBannerID);
632
   END
633
634
   #####################################
635
636
637
   DELIMITER $$
638
   CREATE PROCEDURE AssignBannerID( IN cwid INT(11),
639
    OUT UniqueBannerID VARCHAR(20))
640
641
642
   BEGIN
643
644
645
       DECLARE RoleAcronymStudent VARCHAR(1) DEFAULT 's';
646
       DECLARE RoleAcronymInstructor VARCHAR(3) DEFAULT 'in';
647
648
       SELECT @PersonRole := Role FROM Person WHERE @cwid = CampusWideID;
649
       SELECT @PersonCWID := CampusWideID FROM Person WHERE @cwid = CampusWideI
650
       SELECT @PersonDoB := DoB FROM Person WHERE @cwid = CampusWideID;
651
652
        IF @PersonRole = 'Student' THEN
653
       SELECT Concat(SelectedBannerIDAcronym, PersonCWID, PersonDoB)
654
       INTO UniqueBannerID;
655
656
       ELSEIF @PersonRole = 'Instructor' THEN
657
       SELECT Concat(SelectedBannerIDAcronym, PersonCWID, PersonDoB)
658
       INTO UniqueBannerID;
659
660
       END IF:
661
662
663
       UPDATE Person
664
       SET BannerID = @UniqueBannerID
665
       WHERE @cwid = CampusWideID;
666
667
668
669
   END$$
670
   DELIMITER;
671
672
```