A Data Model and Partial Implementation of a Software System: The Technical Academy Case Study

By

Jeff David

Table of Contents

Table of Figures

1.0 Summary

This is a documentation for the proposed data model for the Technical Academy case study. The proposed model is shown using an entity relationship diagram, drawn using the oracle data modeler tool. The relational diagram is engineered from the entity relationship diagram and the data definition language script is generated. To test the efficiency of this model, data is collected from the List of Names Dataset (Back4App, 2020) and the University of Wolverhampton website (Wlv.ac.uk, 2018). This data is inserted into each table and five queries that exhibit a join, aggregate function, selection, subquery and sorting of data are tested on it. This document is concluded with a short discussion on data governance rules in a university environment like the Technical Academy.

2.0 Introduction

The education sector is the sector of interest for this case study and the potential client business is a university in Central England called The Technical Academy. The Technical Academy offers several courses and has corresponding schools responsible for these courses. The university boasts of competent lecturers and bright students but lacks a reliable record system to hold information about the courses, modules, students, lecturers, and activities surrounding student assessments and time spent teaching a module. Hence, the university has outsourced the development of a data model and the partial implementation of a software system to help resolve this issue. This report shows the design process of the database and its partial implementation using a set of test data (Nassa and Frommholz, 2021).

3.0 Requirement Specification

3.1 Entity Relationship Diagram

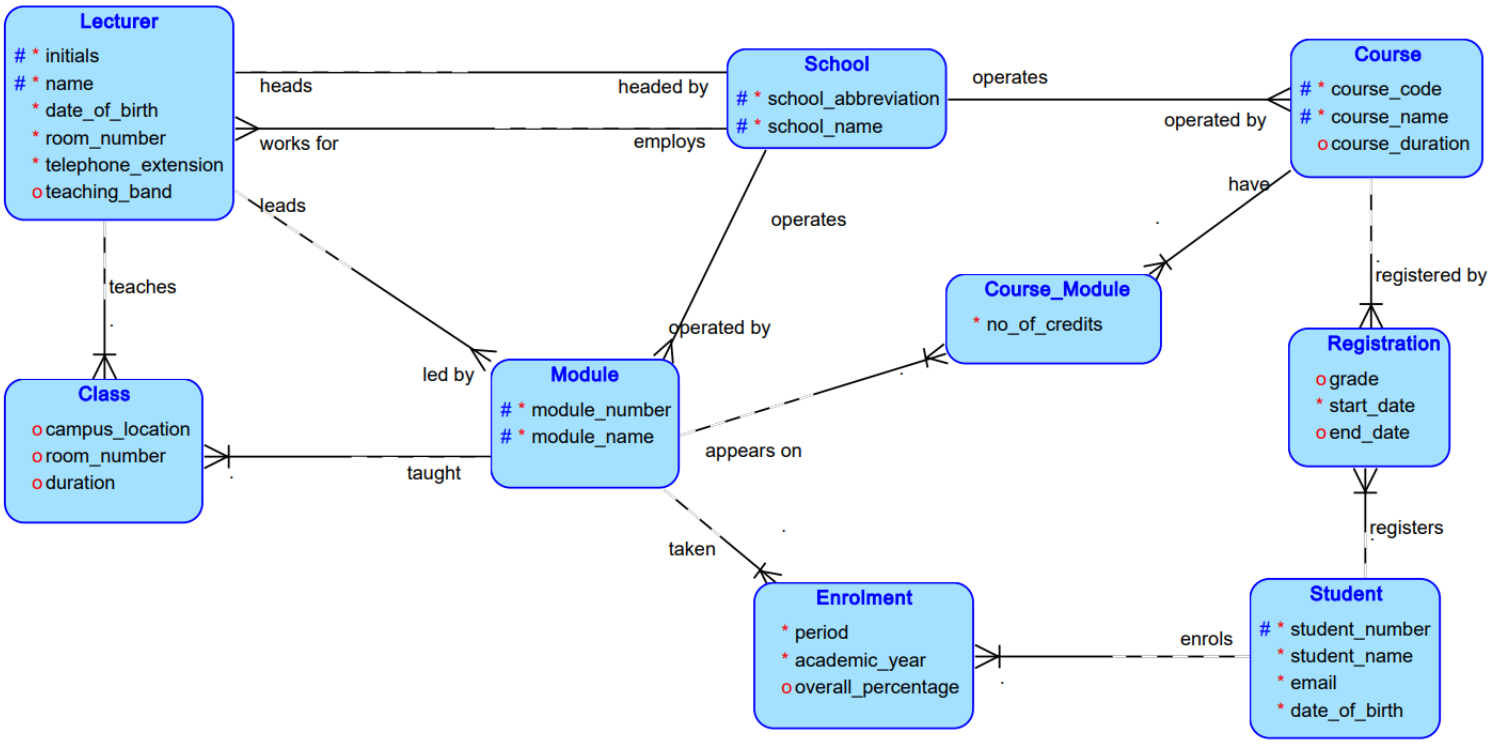

Figure 1: Proposed Entity Relationship Diagram for the Technical Academy

3.1.1 Assumptions

1. Firstly, this design considers only one instance of entity, 'School' in relationship to every other entity.
2. In the relationship between Lecturer and School entity: - Lecturers must work for a particular school but one school cannot employ all the lecturers in the university and a school must be headed by only one lecturer from that same school. Therefore, the school_name and name are made primary keys because they would serve as foreign keys on the lecturer and school table showing the school_name and head_of_school column respectively. Meanwhile, initials and school_abbreviation are primary keys because they are required to be unique.

Lecturer Entity					
Attribute Name	Type	Size	Primary Key	Mandatory	Constraint
initials	VARCHAR	4	Yes	Yes (by default)	-
name	VARCHAR	20	Yes	Yes (by default)	-
date_of_birth	Date	-	No	Yes	-
room_number	CHAR	8	No	Yes	-
telephone_extension	NUMERIC	Precision: 4	No	Yes	-
teaching_band	NUMERIC	-	No	No	List of Ranges: 50 - 500

School Entity					
Attribute Name	Type	Size	Primary Key	Mandatory	Constraint
school_abbreviation	CHAR	5	Yes	Yes (by default)	-
school_name	VARCHAR	20	Yes	Yes (by default)	-

3. In the relationship between School and Course entity: - A school must have courses its responsible for and every course must appear under a school as well. Therefore, the school_name is made a primary key because it would serve as a foreign key on the course table showing the school responsible for the course. Meanwhile, school_abbreviation and course_code are primary keys because they are required to be unique. Also, a domain is set for course_duration; a course length is set to be between 1 and 7.

Course Entity					
Attribute Name	Type	Size	Primary Key	Mandatory	Constraint
course_code	CHAR	5	Yes	Yes (by default)	-
course_name	VARCHAR	30	Yes	Yes (by default)	-
course_duration	NUMERIC	-	No	No	List of Ranges: 1 - 7

4. In the relationship between Course and Student entity: - Because it is a many to many (M:N) relationship within a school, a link entity called “Registration” is created with attributes like grade, start_date and end_date. Grade is not mandatory to fill at the point of registration because this information will only be available at the successful completion of a course. Start_date is mandatory to fill because at the point of registration, there will be a course start_date. End_date is not mandatory to fill because at the point of registration, it is not certain when or if the student will successfully complete the course; extenuating circumstances can affect the end_date.

   A course maybe registered by several students and therefore, the course_name is made a primary key because it would serve as a foreign key on the registration table showing the course name a student registered on. Meanwhile, course_code is a primary key because it is required to be unique.

   A student may register to one of the courses that is operated by a school and the student_number will serve as a foreign key on the registration table identifying the student that registered on the course. The student_number is considered enough to identify a student because it is unique.

Student Entity					
Attribute Name	Type	Size	Primary Key	Mandatory	Constraint
student_number	CHAR	8	Yes	Yes (by default)	-
student_name	VARCHAR	20	No	Yes	-
email	VARCHAR	30	No	Yes	-
date_of_birth	Date	-	No	Yes	-

5. In the relationship between Student and Module entity: - Because it is a many to many (M:N) relationship within a school, a link entity called “Enrolment” is created with attributes like period, academic_year and overall_percentage. Period is mandatory to fill because students need to know the exact semester a particular module will be taken at the point of enrolment. Academic_year is mandatory to fill because it is also important for the students to know the academic year a module will be taken. The overall_percentage is not mandatory to fill at the point of enrolment because this information will only be available at the successful completion of a module.

   A student may enroll on a module and therefore, the student_number will serve as a foreign key on the enrolment table identifying the student that is enrolled on the module. Again, the student_number is considered enough to identify a student because it is unique.

   A module maybe taken by several students and therefore, the module_name is made a primary key because it would serve as a foreign key on the enrolment table showing the module_name the student enrolled on. Meanwhile, module_number is a primary key because it is required to be unique.

Module Entity					
Attribute Name	Type	Size	Primary Key	Mandatory	Constraint
module_number	CHAR	6	Yes	Yes (by default)	-
module_name	VARCHAR	20	Yes	Yes (by default)	-

6. In the relationship between Lecturer and Module entity: - Lecturers may teach several modules and several modules maybe taught by many lecturers. Also, one lecturer may lead several modules and a module must be led by one lecturer. Therefore, it is a many to many (M:N) relationship within a school and a link entity called "Class" is created with attributes like campus_location, room_number and duration. All the three attributes are not mandatory to fill at the start of a semester or course because the timetable will have to be confirmed by the school's student services department for the campus_location, room_number and duration of the class. Also, a domain, campus_location_value is set for campus_location; a class's campus_location is either CHELTENHAM, HEREFORD, or WARWICK.

   A module maybe taught in several classes and therefore, the module_name will serve as a foreign key on the class table identifying the module to be taught in a class. Again, the module_number is a primary key because it is required to be unique.

   A lecturer may teach in several classes and therefore, name will serve as a foreign key on the class table showing the lecturer responsible for a class and the time expected to be spent teach a module in a class. Also, a lecturer may lead several modules and therefore, name will serve as a foreign key on the module table showing the name of the lecturer responsible for leading a module. Meanwhile, initials is a primary key because it is required to be unique.

7. In the relationship between School and Module entity: - A school must operate several modules and a module must be operated by a school. Therefore, school_name will serve as a foreign key on the module table showing the name of the school responsible for a module. Meanwhile, school_abbreviation is a primary key because it is required to be unique.

8. In the relationship between Course and Module entity: - Because it is a many to many (M:N) relationship within a school, a link entity called "Course_Module" is created with no_of_credits as its only attribute. No_of_credits is mandatory to fill because this information should be set at the point of provisioning a new course to be operated by a school.

   A course must have several modules and therefore, the course_name is made a primary key because it would serve as a foreign key on the course_module table showing the name of the course responsible for a module. Meanwhile, course_code is a primary key because it is required to be unique.

   A module may appear on several courses and therefore, the module_name will serve as a foreign key on the course_module table identifying the module that appears on a course. The module_number is a primary key because it is required to be unique.

Class Entity					
Attribute Name	Type	Size	Primary Key	Mandatory	Constraint
campus_location	VARCHAR	10	No	No	List of Values: CHELTENHAM, HEREFORD, WARWICK
room_number	CHAR	8	No	No	-
duration	VARCHAR	10	No	No	-

Course_Module Entity					
Attribute Name	Type	Size	Primary Key	Mandatory	Constraint
no_of_credits	NUMERIC	Precision: 2	No	Yes	-

Registration Entity					
Attribute Name	Type	Size	Primary Key	Mandatory	Constraint
grade	VARCHAR	20	No	No	-
start_date	Date	-	No	Yes	-
end_date	Date	-	No	No	-

Enrolment Entity					
Attribute Name	Type	Size	Primary Key	Mandatory	Constraint
period	VARCHAR	9	No	Yes	-
academic_year	VARCHAR	9	No	Yes	-
overall_percentage	VARCHAR	4	No	No	-

3.2 *Relational Diagram*

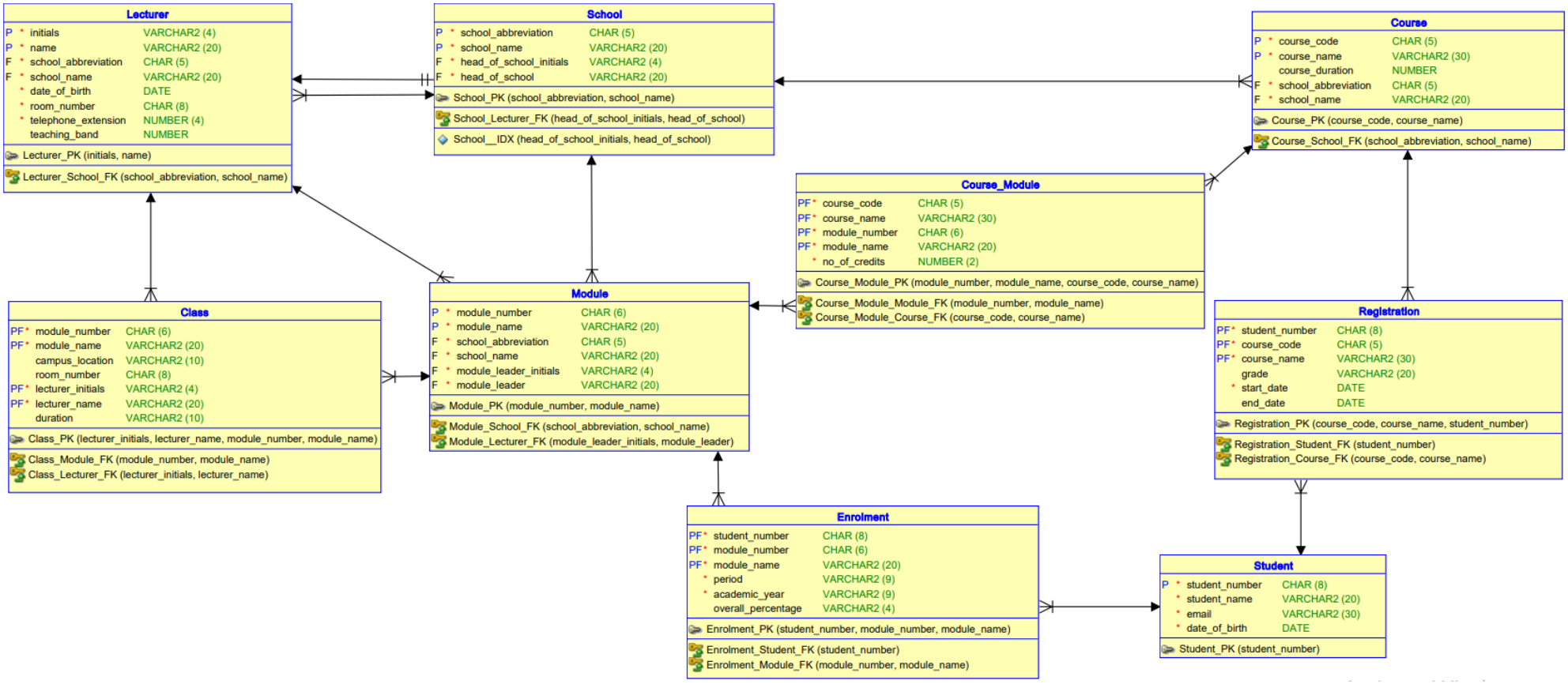

Figure 2: Proposed Relational Diagram for the Technical Academy

3.2.1 Comments

1. Additional domain, EndDate_chk is added to the Registration table; start_date < end_date.
2. The foreign key names are renamed to make meaning on the tables they appear on.
   a. All the prefixes (name of the entity it related to) are removed from the foreign keys like in Enrolment table; Student_student_number is renamed to student_number.

b. On the School table, initials is renamed to head_of_school_initials and name is renamed to head_of_school.

c. On the Module table, Lecturer_initials is renamed to module_leader_initials and Lecturer_name is renamed to module_leader.

3.3 Data Collection and Populating the Database

This report used the List of Names Dataset (Back4App, 2020) to populate the Lecturer and Student tables. The dataset is publicly available and is made up of 250,000 names categorized by gender and letter. While data found on the University of Wolverhampton website (Wlv.ac.uk, 2018) is used to populate the School, Course and Module tables. The University of Wolverhampton is a public university in England. The content of each table is shown below: -

```
SQL> SELECT * FROM LECTURER;

INIT NAME                 SCHOO SCHOOL_NAME          DATE_OF_B ROOM_NUM TELEPHONE_EXTENSION TEACHING_BAND
---- -------------------- ----- -------------------- --------- -------- ------------------- -------------
WIJA WILL JAMES           SOENG ENGINEERING          26-MAY-84 HFRD0034                1011           450
SAJO SAMUEL JOHN          SOMAT MATHS                09-SEP-77 HFRD0201                1111           450
GEFR GEORGE FRED          SOMED MEDICINE             16-FEB-70 CTHM0034                1011           450
ELRI ELIANNA RICHARD      SOPHM PHARMACY             24-JUN-75 CTHM0201                1111           450
CAWI CARLEIGH WILLIAM     SOSCI SCIENCES             03-JAN-80 WRWK0034                1011           450
KABE KATHY BERT           SOART ART                  27-NOV-78 WRWK0201                1111           450
ALDA ALBERT DAVID         SOENG ENGINEERING          30-APR-88 HFRD0035                1012           150
CAHE CARL HENRY           SOENG ENGINEERING          04-FEB-89 HFRD0036                1013           350
SHWA SHAYNA WALTER        SOENG ENGINEERING          05-MAY-87 HFRD0037                1014           200
LAFE LAUREL FREDERICK     SOENG ENGINEERING          06-JUN-83 HFRD0038                1015           350
ANER ANDREW ERNEST        SOMAT MATHS                30-JUN-88 HFRD0039                1016           150
LELO LEE LOUIS            SOMAT MATHS                04-MAR-89 HFRD0040                1016           150
NERU NEVEAH RUFUS         SOMAT MATHS                07-MAY-77 HFRD0041                1018           400
MATO MAKAILA TOM          SOMAT MATHS                08-MAY-80 HFRD0042                1019           200
BEFR BEN FRANCIS          SOMED MEDICINE             17-FEB-87 CTHM0035                1012           150
MALU MARION LUTHER        SOMED MEDICINE             20-MAR-89 CTHM0036                1013           350
KIJU KIANNA JULIUS        SOMED MEDICINE             05-MAY-89 CTHM0037                1014           200
ROLA RORY LAWRENCE        SOMED MEDICINE             06-JUN-69 CTHM0038                1015           430
DAWA DAN WARREN           SOPHM PHARMACY             30-JUN-88 CTHM0039                1016           150
DAAL DANIEL ALFRED        SOPHM PHARMACY             04-MAR-89 CTHM0040                1017           350
HAHA HADASSAH HARVEY      SOPHM PHARMACY             04-MAR-89 CTHM0041                1018           400
AMJI AMARIS JIM           SOPHM PHARMACY             07-MAY-77 CTHM0042                1019           200
LEED LEO EDWARD           SOSCI SCIENCES             17-FEB-87 WRWK0034                1012           150
WIHO WILLIE HOWARD        SOSCI SCIENCES             20-MAR-89 WRWK0035                1013           350
```

```
DAEU DANIA EUGENE         SOSCI SCIENCES             05-MAY-89 WRWK0036                1014          200
JACL JAMIYA CLARENCE      SOSCI SCIENCES             06-JUN-79 WRWK0037                1015          300
CHLE CHESTER LEWIS        SOART ART                  30-JUN-88 WRWK0038                1016          150
IRHE IRA HERMAN           SOART ART                  04-MAR-89 WRWK0039                1017          350
HACH HAILIE CHARLES       SOART ART                  08-MAY-80 WRWK0040                1018          400
EMCL EMELIA CLYDE         SOART ART                  07-MAY-77 WRWK0041                1019          200

30 rows selected.

SQL> SELECT * FROM SCHOOL;

SCHOO SCHOOL_NAME          HEAD HEAD_OF_SCHOOL
----- -------------------- ---- --------------------
SOSCI SCIENCES             CAWI CARLEIGH WILLIAM
SOPHM PHARMACY             ELRI ELIANNA RICHARD
SOMED MEDICINE             GEFR GEORGE FRED
SOART ART                  KABE KATHY BERT
SOMAT MATHS                SAJO SAMUEL JOHN
SOENG ENGINEERING          WIJA WILL JAMES

6 rows selected.

SQL> SELECT * FROM COURSE;

COURS COURSE_NAME                    COURSE_DURATION SCHOO SCHOOL_NAME
----- ------------------------------ --------------- ----- --------------------
BME11 BENG(HONS) MECH ENG                          3 SOENG ENGINEERING
MME21 MENG(HONS) MECH ENG                          1 SOENG ENGINEERING
BCS11 BSC(HONS) COMP SCI                           3 SOMAT MATHS
MCS21 MSC(HONS) COMP SCI                           1 SOMAT MATHS
BMS11 BSC(HONS) MED SCI                            3 SOMED MEDICINE
MMS21 MSC(HONS) MED SCI                            1 SOMED MEDICINE
BPS11 BSC(HONS) PHARM SCI                          3 SOPHM PHARMACY
MPS21 MSC(HONS) PHARM SCI                          1 SOPHM PHARMACY
BCH11 BSC(HONS) CHEM                               3 SOSCI SCIENCES
MCH21 MSC(HONS) CHEM                               1 SOSCI SCIENCES
BFA11 BA(HONS) FINE ART                            3 SOART ART
MFA21 MA(HONS) FINE ART                            1 SOART ART
```

```
12 rows selected.

SQL> SELECT * FROM STUDENT;

STUDENT_ STUDENT_NAME         EMAIL                          DATE_OF_B
-------- -------------------- ------------------------------ ---------
00000001 ANABEL MICHAEL       A.MICHAEL@TECH.AC.UK           30-APR-02
00000002 ALEX FLOYD           A.FLOYD@TECH.AC.UK             05-MAY-00
00000003 PATRICK ALLEN        P.ALLEN@TECH.AC.UK             05-MAY-00
00000004 EARL JACK            E.JACK@TECH.AC.UK              06-JUN-00
00000005 MILTON OSCAR         M.OSCAR@TECH.AC.UK             30-JUN-01
00000006 ROBERT JACOB         R.JACOB@TECH.AC.UK             04-MAR-99
00000007 ELMER PAUL           E.PAUL@TECH.AC.UK              07-MAY-97
00000008 JOE CHARLIE          J.CHARLIE@TECH.AC.UK           08-MAY-98
00000009 PETER ROY            P.ROY@TECH.AC.UK               17-FEB-97
00000010 GUY HERBERT          G.HERBERT@TECH.AC.UK           20-MAR-95
00000011 HUGH WILLIS          H.WILLIS@TECH.AC.UK            05-MAY-96
00000012 BERNARD JESSIE       B.JESSIE@TECH.AC.UK            06-JUN-99
OOOOOO13 EDWIN OLIVER         E.OLIVER@TECH.AC.UK            30-JUN-98
00000014 SIDNEY ISAAC         S.ISAAC@TECH.AC.UK             04-MAR-99
00000015 PHILIP JOSEPH        P.JOSEPH@TECH.AC.UK            08-MAY-94
00000016 CHARLEY EDGAR        C.EDGAR@TECH.AC.UK             07-MAY-94
00000017 RAY LEONARD          R.LEONARD@TECH.AC.UK           30-APR-98
00000018 RALPH CLAUDE         R.CLAUDE@TECH.AC.UK            04-FEB-93
00000019 MARIBEL RAYMOND      M.RAYMOND@TECH.AC.UK           05-MAY-93
00000020 NYASIA JERRY         N.JERRY@TECH.AC.UK             06-JUN-94
00000021 LEIA HORACE          L.HORACE@TECH.AC.UK            30-JUN-97
00000022 ALANNAH ED           A.ED@TECH.AC.UK                04-MAR-96
00000023 GRETCHEN OTTO        G.OTTO@TECH.AC.UK              04-MAY-98
00000024 NAIMA STEPHEN        N.STEPHEN@TECH.AC.UK           07-MAY-93
00000025 LIVIA DAVE           L.DAVE@TECH.AC.UK              17-FEB-91

25 rows selected.

SQL> SELECT * FROM MODULE;

MODULE MODULE_NAME          SCHOO SCHOOL_NAME          MODU MODULE_LEADER
------ -------------------- ----- -------------------- ---- --------------------
4BME11 COMP AIDED DESIGN    SOENG ENGINEERING          CAHE CARL HENRY
```

```
5BME12 THERMODYNAMICS        SOENG ENGINEERING          CAHE CARL HENRY
6MME21 MACHINES DESIGN       SOENG ENGINEERING          LAFE LAUREL FREDERICK
7MME22 SENSORS AND CONTROL   SOENG ENGINEERING          LAFE LAUREL FREDERICK
4BCS11 COMPUTATIONAL MATHS   SOMAT MATHS                LELO LEE LOUIS
5BCS12 NUMERICAL METHODS     SOMAT MATHS                LELO LEE LOUIS
6MCS21 BIG DATA              SOMAT MATHS                NERU NEVEAH RUFUS
7MCS22 COMPLEX SYSTEMS       SOMAT MATHS                NERU NEVEAH RUFUS
4BMS11 BIO OF DISEASE        SOMED MEDICINE             MALU MARION LUTHER
5BMS12 GROWTH AND NUTRITION  SOMED MEDICINE             MALU MARION LUTHER
6MMS21 ADVANCED RESEARCH     SOMED MEDICINE             ROLA RORY LAWRENCE
7MMS22 MEDICAL EDUCATION     SOMED MEDICINE             ROLA RORY LAWRENCE
4BPS11 INTRO TO MICROBIO     SOPHM PHARMACY             DAAL DANIEL ALFRED
5BPS12 THERAPEUTIC PHARM     SOPHM PHARMACY             DAAL DANIEL ALFRED
6MPS21 BIOCHEM PHARM         SOPHM PHARMACY             HAHA HADASSAH HARVEY
7MPS22 HUMAN STRUCTURE       SOPHM PHARMACY             HAHA HADASSAH HARVEY
4BCH11 ORGANIC CHEM          SOSCI SCIENCES             WIHO WILLIE HOWARD
5BCH12 INORGANIC CHEM        SOSCI SCIENCES             WIHO WILLIE HOWARD
6MCH21 ANALYTICAL CHEM       SOSCI SCIENCES             JACL JAMIYA CLARENCE
7MCH22 PHYSICAL CHEM         SOSCI SCIENCES             JACL JAMIYA CLARENCE
4BFA11 FUNDAMENTAL SKILLS    SOART ART                  IRHE IRA HERMAN
5BFA12 WORKSHOP              SOART ART                  IRHE IRA HERMAN
6BFA21 STUDIO CONTEXTS       SOART ART                  HACH HAILIE CHARLES
7BFA22 CREATIVE APPROACHES   SOART ART                  HACH HAILIE CHARLES

24 rows selected.
```

SQL> **SELECT * FROM REGISTRATION;**

```
STUDENT_ COURS COURSE_NAME                    GRADE                START_DAT END_DATE
-------- ----- ------------------------------ -------------------- --------- ---------
00000001 BME11 BENG(HONS) MECH ENG                                 30-SEP-21
00000003 MME21 MENG(HONS) MECH ENG                                 30-SEP-21
00000004 BCS11 BSC(HONS) COMP SCI                                  30-SEP-21
00000005 MCS21 MSC(HONS) COMP SCI                                  30-SEP-21
00000006 MCH21 MSC(HONS) CHEM                                      30-SEP-21
00000002 MPS21 MSC(HONS) PHARM SCI                                 30-SEP-21
00000007 BME11 BENG(HONS) MECH ENG                                 30-SEP-21
00000008 BME11 BENG(HONS) MECH ENG                                 30-SEP-21
00000009 MME21 MENG(HONS) MECH ENG                                 30-SEP-21
```

```
00000010 BCS11 BSC(HONS) COMP SCI                                    30-SEP-21
00000011 BCS11 BSC(HONS) COMP SCI                                    30-SEP-21

11 rows selected.
```

SQL> **SELECT * FROM ENROLMENT;**

```
STUDENT_ MODULE MODULE_NAME          PERIOD    ACADEMIC_ OVER
-------- ------ -------------------- --------- --------- ----
00000001 4BME11 COMP AIDED DESIGN    SEM 1     2021/2022 66%
00000002 6MPS21 BIOCHEM PHARM        SEM 1     2021/2022 70%
00000003 7MME22 SENSORS AND CONTROL  SEM 2     2021/2022
00000004 5BCS12 NUMERICAL METHODS    SEM 2     2021/2022
00000005 6MCS21 BIG DATA             SEM 1     2021/2022
00000006 6MCH21 ANALYTICAL CHEM      SEM 1     2021/2022

6 rows selected.
```

SQL> **SELECT * FROM COURSE_MODULE;**

```
COURS COURSE_NAME                   MODULE MODULE_NAME          NO_OF_CREDITS
----- ----------------------------- ------ -------------------- -------------
BME11 BENG(HONS) MECH ENG           4BME11 COMP AIDED DESIGN              20
MME21 MENG(HONS) MECH ENG           6MME21 MACHINES DESIGN                20
BCS11 BSC(HONS) COMP SCI            5BCS12 NUMERICAL METHODS              20
MCS21 MSC(HONS) COMP SCI            6MCS21 BIG DATA                       20
BMS11 BSC(HONS) MED SCI             5BMS12 GROWTH AND NUTRITION           20
MMS21 MSC(HONS) MED SCI             7MMS22 MEDICAL EDUCATION              20

6 rows selected.
```

SQL> **SELECT * FROM CLASS;**

```
MODULE MODULE_NAME          CAMPUS_LOC ROOM_NUM LECT LECTURER_NAME        DURATION
------ -------------------- ---------- -------- ---- -------------------- ----------
4BME11 COMP AIDED DESIGN    HEREFORD   HFRD1101 ALDA ALBERT DAVID         30
6MMS21 ADVANCED RESEARCH    CHELTENHAM CTHM1101 BEFR BEN FRANCIS          40
5BCS12 NUMERICAL METHODS    HEREFORD   HFRD1120 ANER ANDREW ERNEST        50
7MPS22 HUMAN STRUCTURE      CHELTENHAM CTHM1120 DAWA DAN WARREN           50
```

```
6MCH21 ANALYTICAL CHEM      WARWICK    WRWK1101 LEED LEO EDWARD          45
7BFA22 CREATIVE APPROACHES  WARWICK    WRWK1120 CHLE CHESTER LEWIS       30

6 rows selected.

SQL> SPOOL OFF;
```

3.4 SQL Queries

3.4.1 Query 1

```
SQL> SELECT INITIALS,NAME,CAMPUS_LOCATION,DURATION
     2  FROM LECTURER L, CLASS C
     3  WHERE L.INITIALS = C.LECTURER_INITIALS AND CAMPUS_LOCATION <> 'HEREFORD';
```

Output:

```
INIT NAME                 CAMPUS_LOC DURATION
---- -------------------- ---------- ----------
BEFR BEN FRANCIS          CHELTENHAM 40
CHLE CHESTER LEWIS        WARWICK    30
DAWA DAN WARREN           CHELTENHAM 50
LEED LEO EDWARD           WARWICK    45
```

3.4.2 Query 2

```
SQL> SELECT COUNT(STUDENT_NUMBER)
     2  FROM REGISTRATION
     3  WHERE COURSE_NAME = 'BENG(HONS) MECH ENG';
```

Output:

```
COUNT(STUDENT_NUMBER)
---------------------
                    3
```

3.4.3 Query 3

```
SQL> SELECT INITIALS,NAME,ROOM_NUMBER,TEACHING_BAND
     2  FROM LECTURER
     3  WHERE TEACHING_BAND >= 300;
```

Output:

```
INIT NAME                 ROOM_NUM TEACHING_BAND
---- -------------------- -------- -------------
WIJA WILL JAMES           HFRD0034           450
SAJO SAMUEL JOHN          HFRD0201           450
GEFR GEORGE FRED          CTHM0034           450
ELRI ELIANNA RICHARD      CTHM0201           450
CAWI CARLEIGH WILLIAM     WRWK0034           450
KABE KATHY BERT           WRWK0201           450
CAHE CARL HENRY           HFRD0036           350
LAFE LAUREL FREDERICK     HFRD0038           350
NERU NEVEAH RUFUS         HFRD0041           400
MALU MARION LUTHER        CTHM0036           350
ROLA RORY LAWRENCE        CTHM0038           430
DAAL DANIEL ALFRED        CTHM0040           350
HAHA HADASSAH HARVEY      CTHM0041           400
WIHO WILLIE HOWARD        WRWK0035           350
JACL JAMIYA CLARENCE      WRWK0037           300
IRHE IRA HERMAN           WRWK0039           350
HACH HAILIE CHARLES       WRWK0040           400

17 rows selected.
```

3.4.4 Query 4

```
SQL> SELECT INITIALS,NAME,SCHOOL_ABBREVIATION,ROOM_NUMBER,TEACHING_BAND
     2  FROM LECTURER
     3  WHERE SCHOOL_ABBREVIATION = 'SOPHM' OR SCHOOL_ABBREVIATION = 'SOMED' AND TEACHING_BAND >= '300'
     4  ORDER BY TEACHING_BAND DESC;
```

Output:

```
INIT NAME                 SCHOO ROOM_NUM TEACHING_BAND
---- -------------------- ----- -------- -------------
GEFR GEORGE FRED          SOMED CTHM0034           450
ELRI ELIANNA RICHARD      SOPHM CTHM0201           450
ROLA RORY LAWRENCE        SOMED CTHM0038           430
HAHA HADASSAH HARVEY      SOPHM CTHM0041           400
DAAL DANIEL ALFRED        SOPHM CTHM0040           350
MALU MARION LUTHER        SOMED CTHM0036           350
AMJI AMARIS JIM           SOPHM CTHM0042           200
DAWA DAN WARREN           SOPHM CTHM0039           150

8 rows selected.
```

3.4.5 Query 5

```
SQL> SELECT INITIALS,NAME
     2  FROM LECTURER
     3  WHERE INITIALS NOT IN (SELECT LECTURER_INITIALS FROM CLASS);
```

Output:

```
INIT NAME
---- --------------------
ELRI ELIANNA RICHARD
SAJO SAMUEL JOHN
MATO MAKAILA TOM
DAAL DANIEL ALFRED
HAHA HADASSAH HARVEY
KIJU KIANNA JULIUS
CAHE CARL HENRY
GEFR GEORGE FRED
MALU MARION LUTHER
DAEU DANIA EUGENE
LELO LEE LOUIS
WIJA WILL JAMES
WIHO WILLIE HOWARD
AMJI AMARIS JIM
```

```
HACH HAILIE CHARLES
JACL JAMIYA CLARENCE
LAFE LAUREL FREDERICK
EMCL EMELIA CLYDE
KABE KATHY BERT
CAWI CARLEIGH WILLIAM
NERU NEVEAH RUFUS
ROLA RORY LAWRENCE
SHWA SHAYNA WALTER
IRHE IRA HERMAN

24 rows selected.

SQL> SPOOL OFF;
```

3.5 Report

Data governance is the process of exercising control and authority over managing data. The purpose of managing data is to inflate its quality while decreasing the burden and threat that comes with it (Abraham, Schneider and vom Brocke, 2019). According to Fattah and Setyadi, (2021), universities should examine their data governance systems on a regular basis to keep up with innovation and changes in their environment, as well as adapt to new technology. A set of mechanisms, such as structures, procedures, and relational mechanisms, may be used to implement a data governance framework.

The considerable growth of digital applications, software, and innovations means that education policy and practice will become more dataficated and digitalized (Landri, 2021) . The majority of data oversight in universities are supported by a specified group of members in the form of a committee, council, or working group. The Information Technology departments, as well as Institutional Research, are the second and third biggest groups of staff who support and manage data governance activities (Jim and Chang, 2018). Data governance is an essential component that allows universities to correctly report on their operations, make smart choices, and preserve data for the long term. Some of the issues that arise frequently in universities that do not value data governance are: (1) inconsistencies and mistrust as a result of data silos (2) organizational inadequacies caused by a lack of the flow of information (3) Concerns with compliance and liabilities as a result of factual errors in documentation and a lack of information security protocols (Huron, 2021).

4.0 References

Abraham, R., Schneider, J. and vom Brocke, J. (2019). Data governance: A conceptual framework, structured review, and research agenda. *International Journal of Information Management*, 49, pp.424–438.

Back4App (2020). *List of Names Dataset | Database Hub*. [online] Back4App™. Available at: https://www.back4app.com/database/back4app/list-of-names-dataset [Accessed 5 Dec. 2021].

Fattah, A. and Setyadi, R. (2021). Determinants Effectiveness Information Technology Governance in Higher Education Institution (HEI) using partial least squares structural equation modeling approach (PLS-SEM). *Journal of Physics: Conference Series*, 1807(1), p.012007.

Huron (2021). *Data Governance For Higher Education: How to Turn Institutional Data Into a Competitive Advantage*. [online] Available at: https://www.huronconsultinggroup.com/-/media/Resource-Media-Content/Education/Data-Governance-Higher-Education-Huron-Report.ashx [Accessed 11 Dec. 2021].

Jim, C.K. and Chang, H. (2018). The current state of data governance in higher education. *Proceedings of the Association for Information Science and Technology*, 55(1), pp.198–206.

Landri, P. (2021). To resist, or to align? The enactment of data-based school governance in Italy. *Educational Assessment, Evaluation and Accountability*, 33(3), pp.563–580.

Nassa, S. and Frommholz, I. (2021). *7CI006 Portfolio 2021-2022*. [online] canvas.wlv.ac.uk. Available at: https://canvas.wlv.ac.uk/courses/30704/files/4141668.

Wlv.ac.uk. (2018). *University of Wolverhampton - University of Wolverhampton*. [online] Available at: https://www.wlv.ac.uk/ [Accessed 5 Dec. 2021].

5.0 Appendix – DDL Script

```
-- Generated by Oracle SQL Developer Data Modeler 21.2.0.183.1957
--   at:        2021-12-06 06:52:32 GMT
--   site:      Oracle Database 21c
--   type:      Oracle Database 21c



-- predefined type, no DDL - MDSYS.SDO_GEOMETRY

-- predefined type, no DDL - XMLTYPE

CREATE TABLE class (
    module_number     CHAR(6) NOT NULL,
    module_name       VARCHAR2(20) NOT NULL,
    campus_location   VARCHAR2(10),
    room_number       CHAR(8),
    lecturer_initials VARCHAR2(4) NOT NULL,
    lecturer_name     VARCHAR2(20) NOT NULL,
    duration          VARCHAR2(10)
);

ALTER TABLE class
    ADD CONSTRAINT campus_location_value CHECK ( campus_location IN ( '',
'CHELTENHAM', 'HEREFORD', 'WARWICK' ) );

ALTER TABLE class
    ADD CONSTRAINT class_pk PRIMARY KEY ( lecturer_initials,
                                          lecturer_name,
                                          module_number,
                                          module_name );

CREATE TABLE course (
    course_code         CHAR(5) NOT NULL,
    course_name         VARCHAR2(30) NOT NULL,
    course_duration     NUMBER,
    school_abbreviation CHAR(5) NOT NULL,
    school_name         VARCHAR2(20) NOT NULL
);

ALTER TABLE course
    ADD CONSTRAINT course_duration_range CHECK ( course_duration BETWEEN 1 AND 7
);

ALTER TABLE course ADD CONSTRAINT course_pk PRIMARY KEY ( course_code,
                                                          course_name );

CREATE TABLE course_module (
    course_code   CHAR(5) NOT NULL,
    course_name   VARCHAR2(30) NOT NULL,
    module_number CHAR(6) NOT NULL,
    module_name   VARCHAR2(20) NOT NULL,
    no_of_credits NUMBER(2) NOT NULL
```

```
);

ALTER TABLE course_module
    ADD CONSTRAINT course_module_pk PRIMARY KEY ( module_number,
                                                  module_name,
                                                  course_code,
                                                  course_name );

CREATE TABLE enrolment (
    student_number     CHAR(8) NOT NULL,
    module_number      CHAR(6) NOT NULL,
    module_name        VARCHAR2(20) NOT NULL,
    period             VARCHAR2(9) NOT NULL,
    academic_year      VARCHAR2(9) NOT NULL,
    overall_percentage VARCHAR2(4)
);

ALTER TABLE enrolment
    ADD CONSTRAINT enrolment_pk PRIMARY KEY ( student_number,
                                              module_number,
                                              module_name );

CREATE TABLE lecturer (
    initials            VARCHAR2(4) NOT NULL,
    name                VARCHAR2(20) NOT NULL,
    school_abbreviation CHAR(5) NOT NULL,
    school_name         VARCHAR2(20) NOT NULL,
    date_of_birth       DATE NOT NULL,
    room_number         CHAR(8) NOT NULL,
    telephone_extension NUMBER(4) NOT NULL,
    teaching_band       NUMBER
);

ALTER TABLE lecturer
    ADD CONSTRAINT teaching_band_range CHECK ( teaching_band BETWEEN 50 AND 500
);

ALTER TABLE lecturer ADD CONSTRAINT lecturer_pk PRIMARY KEY ( initials,
                                                             name );

CREATE TABLE module (
    module_number          CHAR(6) NOT NULL,
    module_name            VARCHAR2(20) NOT NULL,
    school_abbreviation    CHAR(5) NOT NULL,
    school_name            VARCHAR2(20) NOT NULL,
    module_leader_initials VARCHAR2(4) NOT NULL,
    module_leader          VARCHAR2(20) NOT NULL
);

ALTER TABLE module ADD CONSTRAINT module_pk PRIMARY KEY ( module_number,
                                                          module_name );

CREATE TABLE registration (
    student_number CHAR(8) NOT NULL,
```

```
    course_code    CHAR(5) NOT NULL,
    course_name    VARCHAR2(30) NOT NULL,
    grade          VARCHAR2(20),
    start_date     DATE NOT NULL,
    end_date       DATE
);

ALTER TABLE registration ADD CONSTRAINT enddate_chk CHECK ( start_date < end_date
);

ALTER TABLE registration
    ADD CONSTRAINT registration_pk PRIMARY KEY ( course_code,
                                                 course_name,
                                                 student_number );

CREATE TABLE school (
    school_abbreviation     CHAR(5) NOT NULL,
    school_name             VARCHAR2(20) NOT NULL,
    head_of_school_initials VARCHAR2(4) NOT NULL,
    head_of_school          VARCHAR2(20) NOT NULL
);

CREATE UNIQUE INDEX school__idx ON
    school (
        head_of_school_initials
    ASC,
        head_of_school
    ASC );

ALTER TABLE school ADD CONSTRAINT school_pk PRIMARY KEY ( school_abbreviation,
                                                          school_name );

CREATE TABLE student (
    student_number CHAR(8) NOT NULL,
    student_name   VARCHAR2(20) NOT NULL,
    email          VARCHAR2(30) NOT NULL,
    date_of_birth  DATE NOT NULL
);

ALTER TABLE student ADD CONSTRAINT student_pk PRIMARY KEY ( student_number );

ALTER TABLE class
    ADD CONSTRAINT class_lecturer_fk FOREIGN KEY ( lecturer_initials,
                                                   lecturer_name )
        REFERENCES lecturer ( initials,
                              name );

ALTER TABLE class
    ADD CONSTRAINT class_module_fk FOREIGN KEY ( module_number,
                                                 module_name )
        REFERENCES module ( module_number,
                            module_name );

ALTER TABLE course_module
```

```
    ADD CONSTRAINT course_module_course_fk FOREIGN KEY ( course_code,
                                                         course_name )
        REFERENCES course ( course_code,
                            course_name );

ALTER TABLE course_module
    ADD CONSTRAINT course_module_module_fk FOREIGN KEY ( module_number,
                                                         module_name )
        REFERENCES module ( module_number,
                            module_name );

ALTER TABLE course
    ADD CONSTRAINT course_school_fk FOREIGN KEY ( school_abbreviation,
                                                  school_name )
        REFERENCES school ( school_abbreviation,
                            school_name );

ALTER TABLE enrolment
    ADD CONSTRAINT enrolment_module_fk FOREIGN KEY ( module_number,
                                                     module_name )
        REFERENCES module ( module_number,
                            module_name );

ALTER TABLE enrolment
    ADD CONSTRAINT enrolment_student_fk FOREIGN KEY ( student_number )
        REFERENCES student ( student_number );

ALTER TABLE lecturer
    ADD CONSTRAINT lecturer_school_fk FOREIGN KEY ( school_abbreviation,
                                                    school_name )
        REFERENCES school ( school_abbreviation,
                            school_name );

ALTER TABLE module
    ADD CONSTRAINT module_lecturer_fk FOREIGN KEY ( module_leader_initials,
                                                    module_leader )
        REFERENCES lecturer ( initials,
                              name );

ALTER TABLE module
    ADD CONSTRAINT module_school_fk FOREIGN KEY ( school_abbreviation,
                                                  school_name )
        REFERENCES school ( school_abbreviation,
                            school_name );

ALTER TABLE registration
    ADD CONSTRAINT registration_course_fk FOREIGN KEY ( course_code,
                                                        course_name )
        REFERENCES course ( course_code,
                            course_name );

ALTER TABLE registration
    ADD CONSTRAINT registration_student_fk FOREIGN KEY ( student_number )
        REFERENCES student ( student_number );
```

```
ALTER TABLE school
    ADD CONSTRAINT school_lecturer_fk FOREIGN KEY ( head_of_school_initials,
                                                    head_of_school )
        REFERENCES lecturer ( initials,
                              name );



-- Oracle SQL Developer Data Modeler Summary Report:
--
-- CREATE TABLE                             9
-- CREATE INDEX                             1
-- ALTER TABLE                             26
-- CREATE VIEW                              0
-- ALTER VIEW                               0
-- CREATE PACKAGE                           0
-- CREATE PACKAGE BODY                      0
-- CREATE PROCEDURE                         0
-- CREATE FUNCTION                          0
-- CREATE TRIGGER                           0
-- ALTER TRIGGER                            0
-- CREATE COLLECTION TYPE                   0
-- CREATE STRUCTURED TYPE                   0
-- CREATE STRUCTURED TYPE BODY              0
-- CREATE CLUSTER                           0
-- CREATE CONTEXT                           0
-- CREATE DATABASE                          0
-- CREATE DIMENSION                         0
-- CREATE DIRECTORY                         0
-- CREATE DISK GROUP                        0
-- CREATE ROLE                              0
-- CREATE ROLLBACK SEGMENT                  0
-- CREATE SEQUENCE                          0
-- CREATE MATERIALIZED VIEW                 0
-- CREATE MATERIALIZED VIEW LOG             0
-- CREATE SYNONYM                           0
-- CREATE TABLESPACE                        0
-- CREATE USER                              0
--
-- DROP TABLESPACE                          0
-- DROP DATABASE                            0
--
-- REDACTION POLICY                         0
--
-- ORDS DROP SCHEMA                         0
-- ORDS ENABLE SCHEMA                       0
-- ORDS ENABLE OBJECT                       0
--
-- ERRORS                                   0
-- WARNINGS                                 0
```