**Title**

ASSIGNMENT 2A

**Harry Newton**

n10133810

**Adam Tralka**

n10133542

Tutor

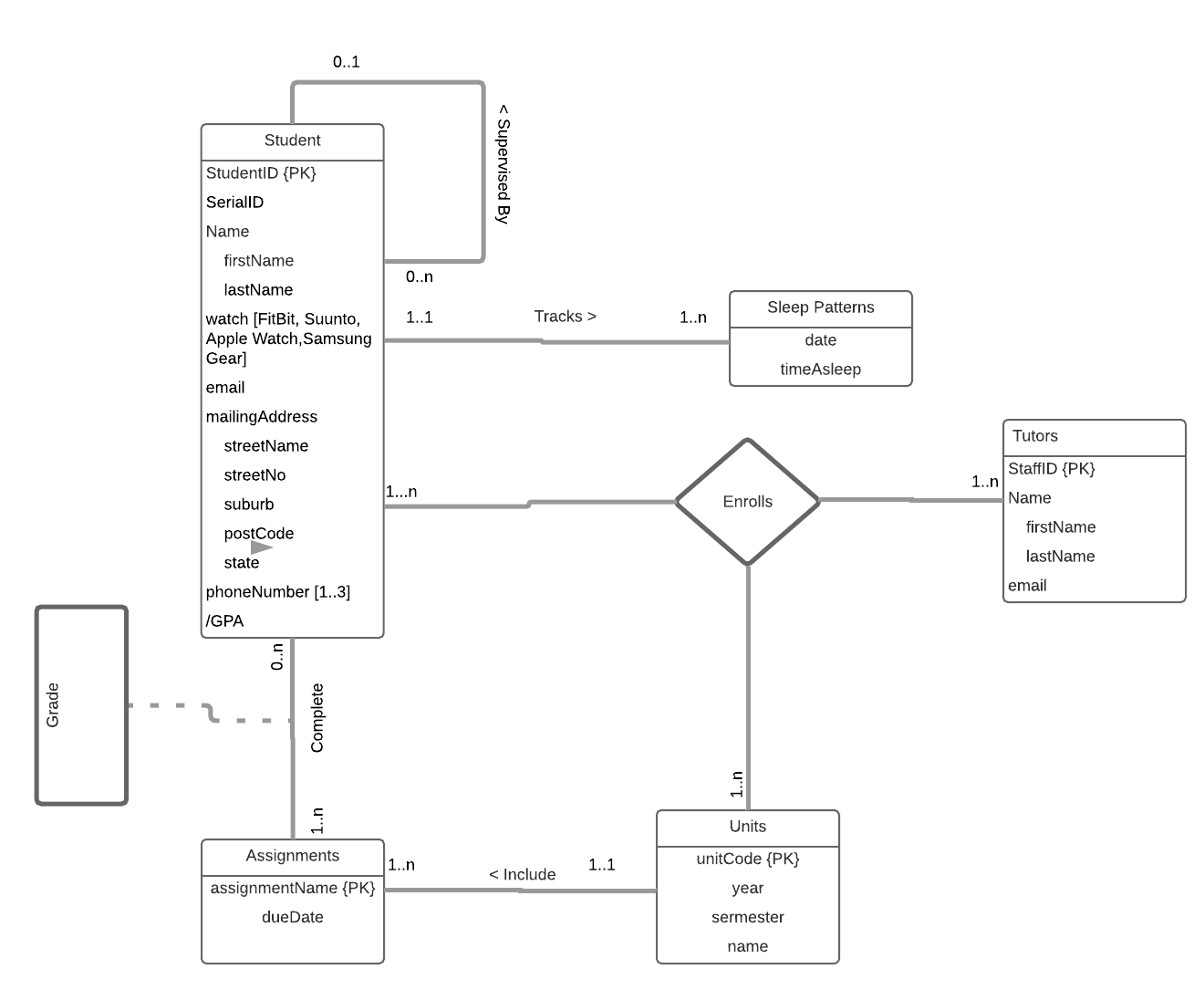
**Jake Bradford**

IFB130 Database Management

April 2018

# Task 1

The database for *Pulse Learning* is modelled in Figure 1. The diagram is based on the information provided by Pulse Learning.



**SAMPLE ONLY TO DEMONSTRATE THE FORMAT EXPECTED**

Figure 1 Entity Relationship Diagram of the *Pulse Learning* Database

**Assumptions:**

* We assumed grades is an attribute of the relationship between the student relation and the assignment relation.
* It was assumed that each student would be enrolled in at least one unit
* Since it is assumed that each student is in at least one unit, it can be assumed that they are assigned at least one assignment.
* It can be assumed that since each student is enrolled in a unit, each student has a tutor.
* It is assumed that a phone number is acquired
* It is assumed that the student is in Australia

# Task 2

The relational model of Pulse Learning database is modelled below. The model is derived from the use case and the ERD in Figure **Deriving Entity types**

Student( studentID, serialID, firstName, lastName, watch, email, streetName, streetNo, suburb, postCode, state, phoneNumber, buddy, /GPA)

sleepPatterns(date,timeAsleep)

Tutors(staffID, firstName, lastName, email)

Units(unitCode, year, semester, name)

Assignments(dueDate, assignmentName)

Grade(studentID, serialID, assignmentName)

unitTutors(StaffID,unitCode)

tutorsStudents(StaffID,StudentID)

studentUnits(StudentID, unitCode)

assignmentCompleted(studentID,assignmentName,grade)

studentPhoneNumbers(studentID, number1, number2, number3)

## Deriving Relationship types

Tracks was derived by merging student and Sleep Patterns because student is a parent of sleep

Completes was derived because the cardinality between students and assignments was a many to many relationship

Include was derived because the cardinality between units and assignments

The supervised by relationship adds the buddyID as a foreign key to the student realtion which points to the studentID. This is because the buddy is the child of the student being assisted.

Enrolls is a ternary relationship which adds the primary key of each relationship participating as a foreign key to a new relation called Enrolment

## Relational Model

Student( studentID, serialID, firstName, lastName, watch, email, streetName, streetNo, suburb, postCode, phoneNumber, state, buddy, /GPA)

sleepPatterns(date,timeAsleep)

Tutors(staffID, firstName, lastName, email)

Units(unitCode, year, semester, name)

Assignments(dueDate, assignmentName)

Grade(studentID, serialID, assignmentName)

unitTutors(StaffID,unitCode)

tutorsStudents(StaffID,StudentID)

studentUnits(StudentID, unitCode)

assignmentCompleted(studentID,assignmentName,grade)

studentPhoneNumbers(studentID, number1, number2, number3)

Completes(grade)

include(unitCode)

Tracks(studentID, date, timeAsleep)

supervisedBy(studentID)

### Foreign Keys

* Sleep Patterns (studentID) is dependent on student (studentID)
* Assignment(unitCode) is despendent on Units(unitCode)
* studentUnits(studentID,unitCode) is dependent on Units (unitCode) amd Tutor (staffID)
* unitTutors( staffID, unitCode) is dependent on Tutor (staffID) and Units (unitCode)
* tutorStudent (staffID, unitCode) is dependent on Tutor (staffIF) and Units (unitCode)

### Other Constraints

* The postCode attribute in students can only be between 0000 9999
* The type of watch need to be Fitbit, Apple, Samsung and Suunto
* the state attribute in students is limited to QLD, NSW, WA, SA, NT and TA

# Task 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table |  |  |  |  |
| Attribute | Attribute | Attribute | Attribute | Attribute |
| value | value | Value | Value | Value |
| Value | Value | Value | value | Value |
| Value | Value | value | Value | Value |
| value | Value | value | value | value |

**…**

* A delete operation that would run successfully

Delete the row:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| value | Value | value | value | value |

From Relation

* An update operation that would run successfully

Update the row:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| value | Value | value | value | value |

From Relation to

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| value | Value | value | value | value |

* An update operation that would not run successfully

Update the row:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| value | Value | value | value | value |

From Relation to

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| value | Value | value | value | value |

* An insert operation that would not run successfully

Insert the row:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| value | Value | value | value | value |

Into Relation

**…**

# Task 4

**Student Relation:**

This relation is not in any normal form, due to the fact tha there are multiple values within the unit cells, this is considered 0NF. A 1NF relation is when the intersection of each row and column contains one value.

**Tutor Relation:**

A 2NF relation is described as fulfilling 1NF requirements as well as every non-primary key attribute is fully functional and dependent on the primary key.

In this relation, the primary key is a composite value made up of the studentID, unitCode and tutorID. The tutors first and last name are not fully functionally dependent on the whole primary key. Due to the fact that there exists attributes which are not fully funcitonally dependent on the primary key, the relation is considered 1NF.

**Coordinator Relation:**

A 3NF is considered to fulfil requirements of a 1NF and 2NF relation, as well as the non-primary key attribute is transitively dependent on the primary key.

In the example given, the primary key is the unit code and all the attributes are dependent on the unit code apart from the coordinator name which depends upon the coordinatorID. This is considered a transitive dependency, which means this is in 2NF form.