

## COSC344 Assignment 2

Team: Group 1A

Leader: George Churton (7452237)

Members:

Calvin Pang (2237621),

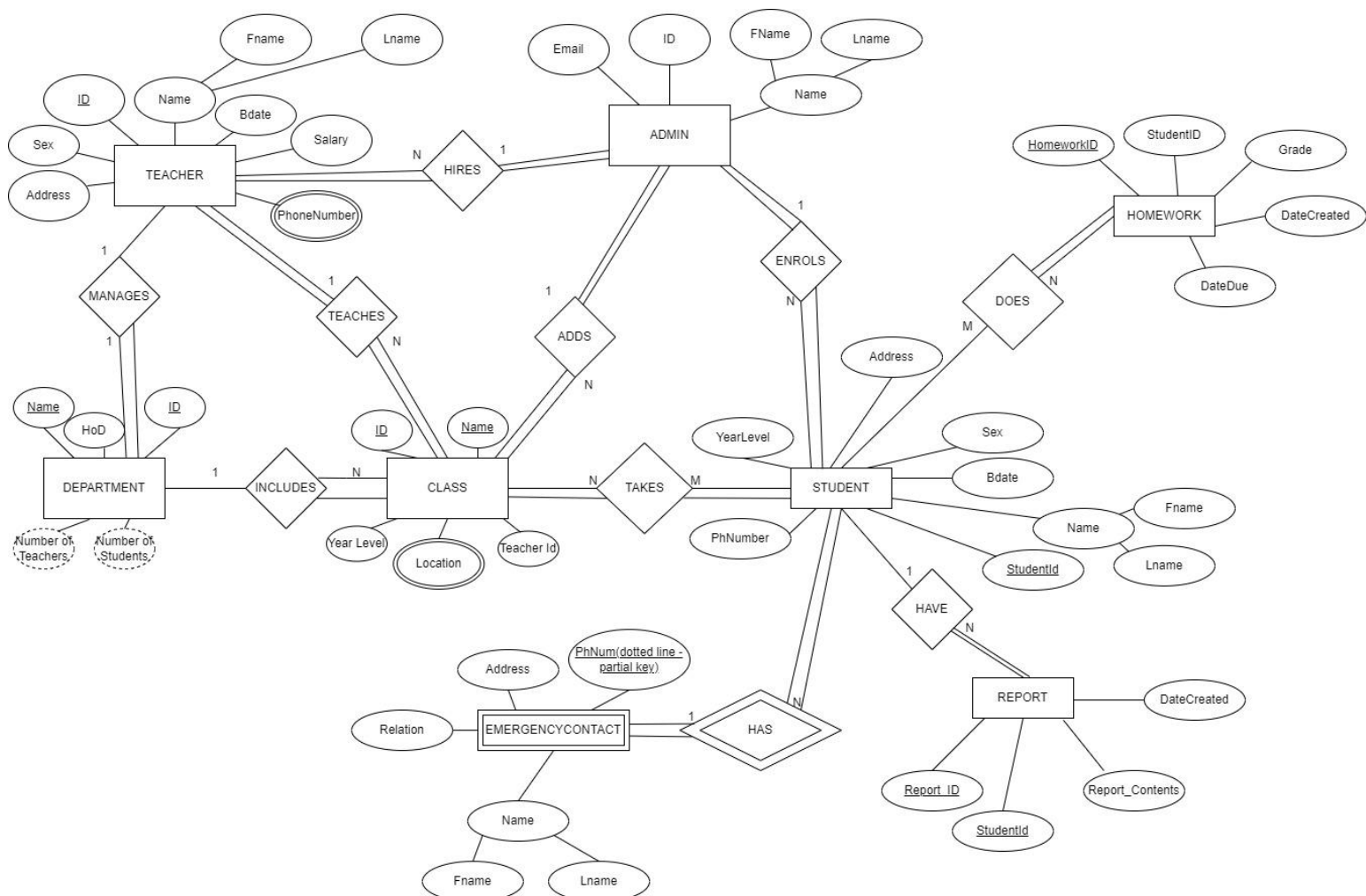
Ben Scobie (5683357),

Oliver Griffin (6262235)

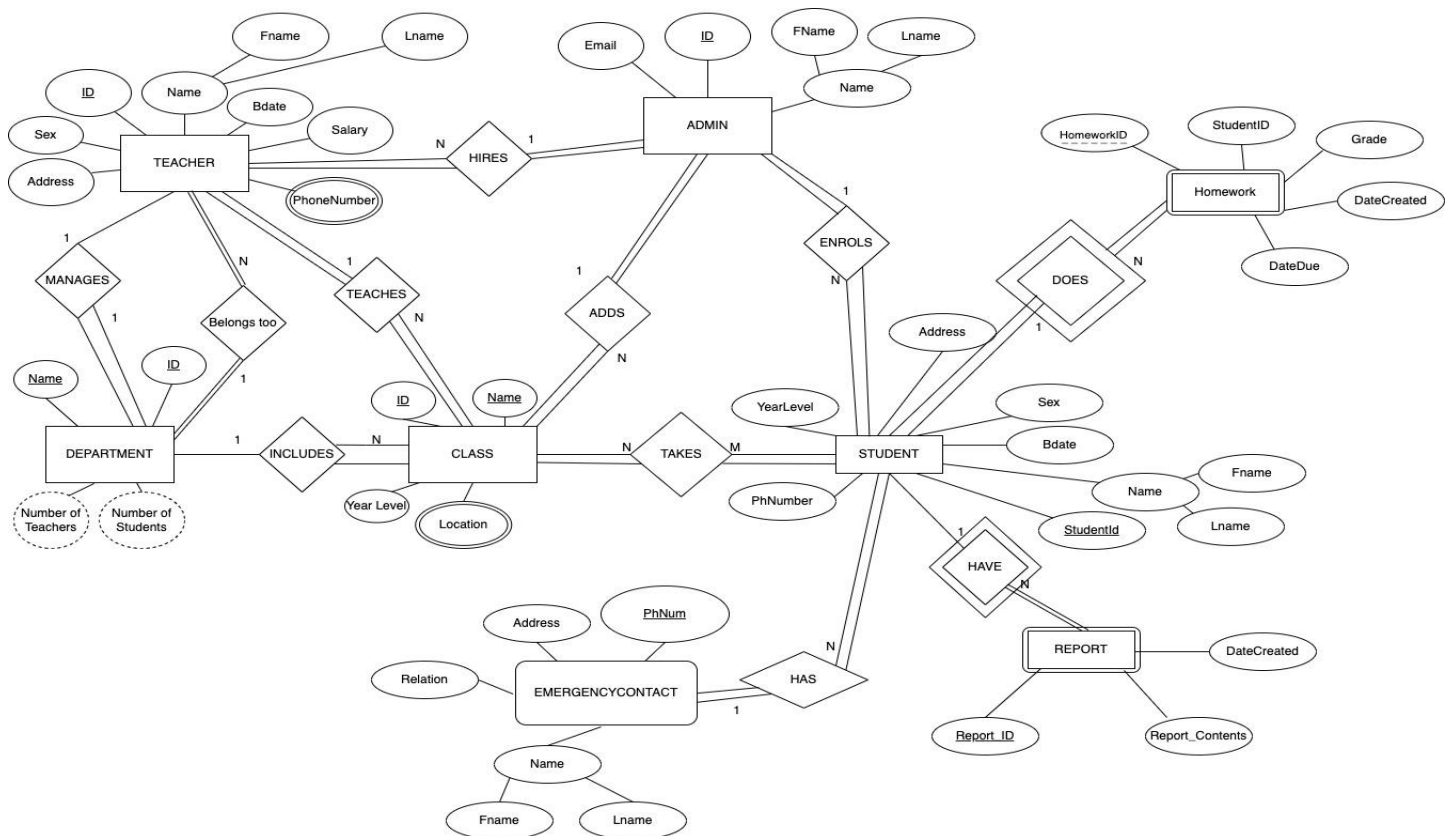
For our project, we are modeling a High School. "Dunedin Grammar". For this school, we are going to keep track of the database's key entities, Students, Teachers, Department, Class, Admin, Report, Homework, Report, and Emergency contacts. These are going to be the foundation of how we are going to model our project. A more in-depth breakdown of these key entities is as follows:

### 1. ER Diagram

Original Diagram:



## Revised ER Diagram:



Once we revisited the original ERD of our school system we recognised some slight changes that we can implement to make our ERD more efficient.

The first update we did was in our 'Class' domain. We noticed that it contained an irrelevant attribute, which was the 'Teacher\_ID' foreign key attribute. This is because it is already represented by the 'Teaches' relationship which is total participation.

Another update we recognised was that in our original ERD we had an unnecessary attribute in our 'Department' domain. In this domain, we had the 'HoD' (Head of Department) attribute, which in our revisited version this attribute was removed because we were already recognising the teacher by the 'Manages' relationship full participation type.

As we continued to recognise that we were doing unnecessary extra work we realised that we had also made the same mistake in our 'Report' domain, where we included another unnecessary attribute 'StudentID'. We also noticed that we could make the 'Report' domain a weak entity type that can be identified by the 'Student' and alongside this to distinguish the reports with the unique identifier 'Report\_ID'.

In our previous report, we talked about a 'Belongs to' relationship that would be used to place teachers in their given department e.g. Biology teachers will belong to the science department. We got carried away focusing on our 'Manages' relationship and forgot to

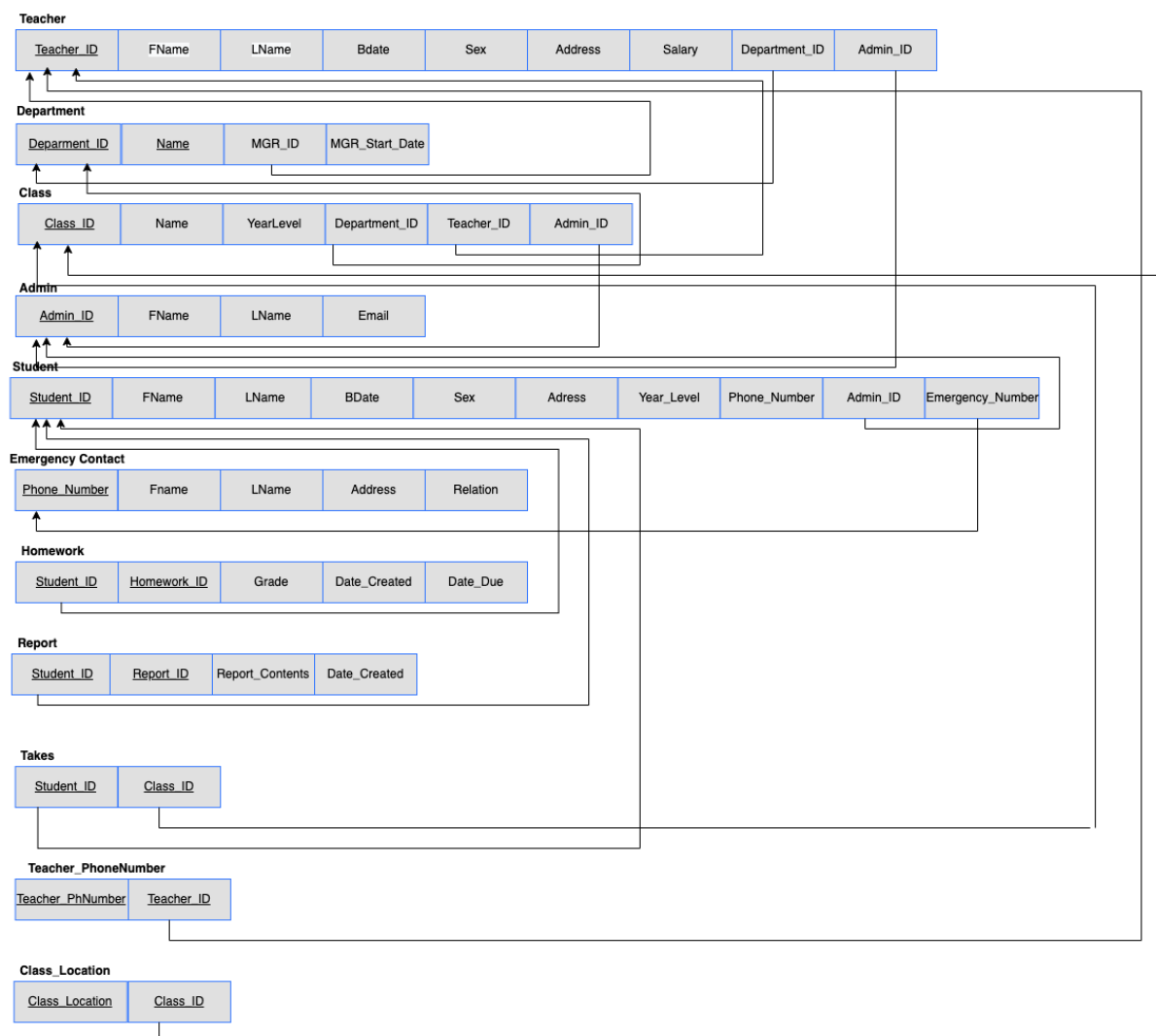
implement the 'Belongs to' relationship that all the none manages needed to be placed into. In our revised ERD we added this 'Belongs to' relationship with full participation as departments must have teachers present and all teachers must belong to one department each.

Another change that we added to our updated ERD diagram is that we did not need to have 'Emergency Contact' as a weak entity type because 'PhNum' will be the key attribute used.

For the 'Homework' domain and 'Does' relationship, we originally had it modelled as full participation. In our revised ERD we change it so that now it is a weak entity type so that each piece of homework can be uniquely identified with the 'StudentID' from the student domain. This is still going to allow us to have the 1:N relationship for the 'Does' relationship.

Overall with these implemented changes, we believe we now have a good foundation to complete our relationship schema and normalisation. We are going to be revisiting our ERD constantly to model our database and system.

## 2. Relational Schema



### 3. Normalisation

Normalisation is done step by step as follows:

#### Step 1: 1NF

Class Location and Teacher phone number already in 1NF

- Used best approach by removing class location and giving it a separate relation with primary key of class and the PK of new relation {Class\_Location#, Class\_ID#}
- Used best approach by removing teacher phone number and giving it a separate relation with primary key of teacher and the PK of new relation {Teacher\_PhNumber#, Teacher\_ID#}

Teacher, Department, Class, Admin, Student, Emergency Contact, Homework, Report and Takes are already in 1NF as all attribute values are atomic and consist of a single value from the domain of the attribute.

#### Step 2: 2NF

Teacher, Class, Admin, Student and Emergency Contact are already in 2NF. We know this as the primary key for these relations to be made of only one attribute and they are also 1NF. Thus, they are fully functionally dependent on the primary key.

Takes, Teacher\_PhoneNumber and Class\_Location are also already in 2NF as there are no non-prime attributes present.

Homework is not in 2NF as Date\_Created and Date\_Due is only dependent on Homework\_ID. To get it into 2NF, we use only Homework\_ID as the primary key as opposed to {Homework\_ID#, Student\_ID#}. Hence, we get the following relation in 2NF  
HOMEWORK{Homework\_ID#, Date\_Created, Date\_Due} and  
STUDENT\_HOMEWORK{HomeworkID#, Student\_ID#}.

Department is not in 2NF as MGR\_ID and MGR\_Start\_Date is only dependent on Department\_ID. To get it into 2NF, we use only Department\_ID as the primary key as opposed to {Department\_ID#, Name#}. Hence, we get the following relation in 2NF  
DEPARTMENT{Department\_ID#, MGR\_ID, MGR\_Start\_Date} and  
DEPARTMENT\_NAME{Department\_ID#, Name#}.

Report is not in 2NF as Report\_Contents and Date\_Created is only dependent on Report\_ID. To get it into 2NF, we use only Report\_ID as the primary key as opposed to {Student\_ID#, Report\_ID#}. Hence, we get the following relation in 2NF  
REPORT{Report\_ID#, Report\_Contents, Date\_Created} and  
STUDENT\_REPORT{Report\_ID#, Student\_ID#}.

#### Step 3: 3NF

Class, Admin, Student, Emergency Contact, Homework, Report, Takes, Teacher\_PhoneNumber and Class\_Location are already in 3NF. For all of these listed relations, their associated attributes are not transitively dependent on the primary key.

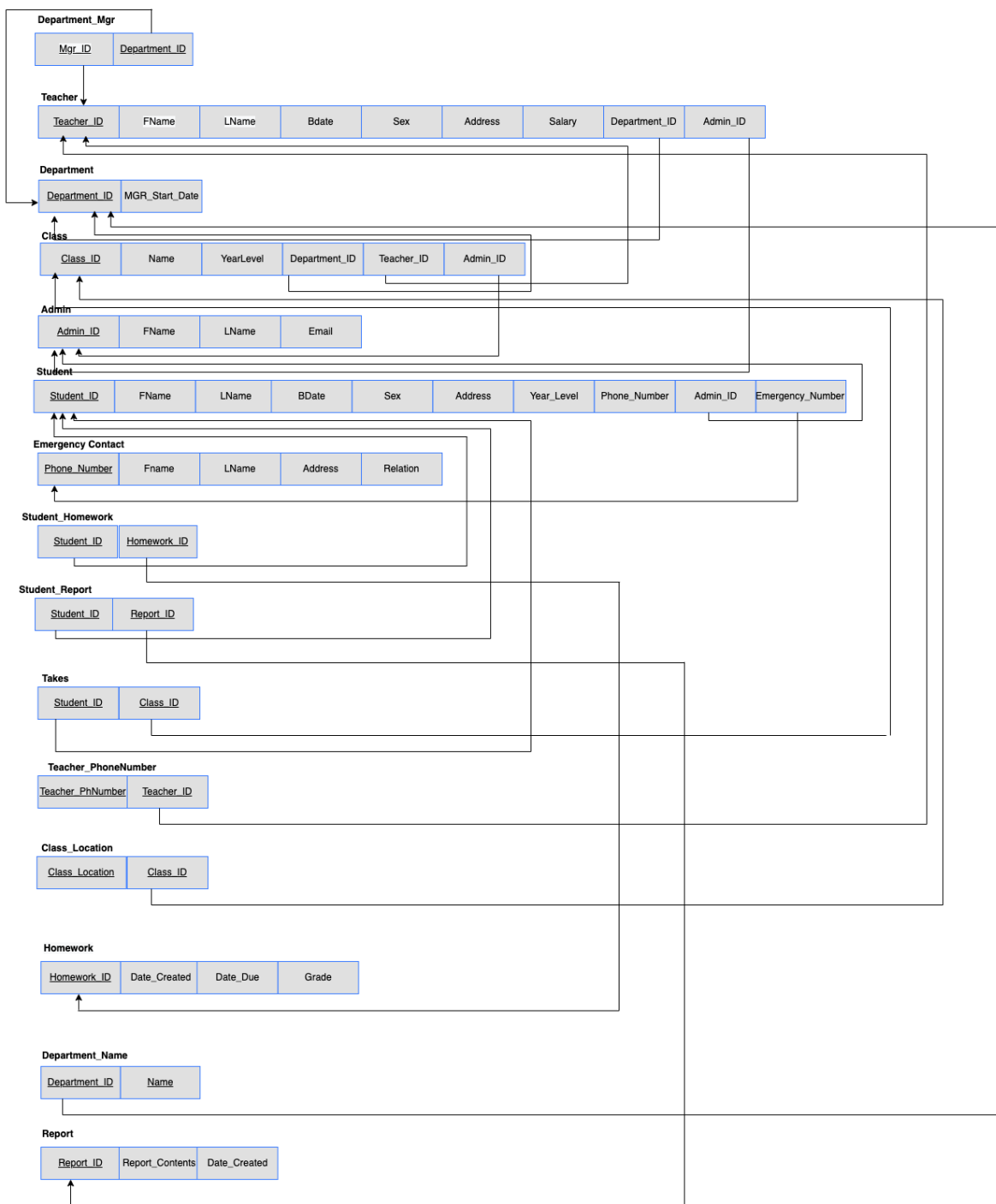
Therefore, since the relations listed above are in 2NF and no non-prime attributes are transitively dependent on the primary key, they are in 3NF.

Department was not in 3NF. MGR\_Start\_Date was transitively dependent on MGR\_ID. To get to 3NF, we remove the transitive dependencies into a new relation called Department\_MGR. Hence, we get the following relation in 3NF  
 DEPARTMENT{Department\_ID#, MGR\_Start\_Date} and  
 DEPARTMENT\_MGR{Department\_ID#, Mgr\_ID#}.

#### Step 4: BCNF

All of our relations are already in BCNF form, as they are in 3NF and each relationship's functional dependency is derived by a superkey. Thus one attribute is non prime, and the other is prime-- this is the result of steps 1-3.

#### Revised Relational Schema:



## 5. Load.sql

```
DROP TABLE Department_Name;
DROP TABLE Department_Mgr;
DROP TABLE Teacher_PhoneNum;
DROP TABLE Class_Location;
DROP TABLE Takes;
DROP TABLE Class;
DROP TABLE Department;
DROP TABLE Teacher;
DROP TABLE Student_Homework;
DROP TABLE Homework;
DROP TABLE Student_Report;
DROP TABLE report;
DROP TABLE Student;
DROP TABLE Emergency_Contact;
DROP TABLE Admin;
```

```
CREATE TABLE Department (
    Department_ID CHAR(5) NOT NULL PRIMARY KEY,
    Mgr_Start_Date DATE NOT NULL
);
```

```
CREATE TABLE Department_Name (
    Department_ID CHAR(5) NOT NULL,
    Name VARCHAR2(20) NOT NULL,
    CONSTRAINT FK_DeptID FOREIGN KEY (Department_ID) REFERENCES
        Department(Department_ID),
    CONSTRAINT PK_Department_Name PRIMARY KEY (Department_ID, Name)
);
```

```
CREATE TABLE Admin (
    Admin_ID CHAR(5) NOT NULL PRIMARY KEY,
    FName VARCHAR2(15) NOT NULL,
    LName VARCHAR2(15) NOT NULL,
    Email VARCHAR2(30) NOT NULL
);
```

```
CREATE TABLE Teacher (
    Teacher_ID CHAR(5) NOT NULL PRIMARY KEY,
    FName VARCHAR2(15) NOT NULL,
    LName VARCHAR2(15) NOT NULL,
    Bdate DATE NOT NULL,
    Sex CHAR NOT NULL,
    Address VARCHAR2(50) NOT NULL,
    Salary NUMBER(8,2) NOT NULL,
    Department_ID CHAR(5),
    Admin_ID CHAR(5),
```

```

        CONSTRAINT FK_TeacherAdmin FOREIGN KEY (Admin_ID) REFERENCES
            Admin(Admin_ID)
    );

CREATE TABLE Department_Mgr (
    Department_ID CHAR(5) NOT NULL,
    Mgr_ID CHAR(5) NOT NULL,
    CONSTRAINT FK_DID FOREIGN KEY (Department_ID) REFERENCES
        Department(Department_ID),
    CONSTRAINT FK_MgrID FOREIGN KEY (Mgr_ID) REFERENCES
        Teacher(Teacher_ID),
    CONSTRAINT PK_Dept PRIMARY KEY (Department_ID, Mgr_ID)
);

CREATE TABLE Teacher_PhoneNum (
    Teacher_ID CHAR(5) NOT NULL,
    Teacher_PhNum INT NOT NULL,
    CONSTRAINT FK_TeacherID FOREIGN KEY (Teacher_ID) REFERENCES
        Teacher(Teacher_ID),
    CONSTRAINT PK_Teacher_PhNum PRIMARY KEY (Teacher_ID,
        Teacher_PhNum)
);

CREATE TABLE Class (
    Class_ID CHAR(5) NOT NULL PRIMARY KEY,
    Name VARCHAR2(15) NOT NULL,
    YearLevel INT NOT NULL,
    Department_ID CHAR(5),
    Teacher_ID CHAR(5) ,
    Admin_ID CHAR(5),
    CONSTRAINT FK_ClassDept FOREIGN KEY (Department_ID) REFERENCES
        Department(Department_ID),
    CONSTRAINT FK_ClassTeacher FOREIGN KEY (Teacher_ID) REFERENCES
        Teacher(Teacher_ID),
    CONSTRAINT FK_ClassAdmin FOREIGN KEY (Admin_ID) REFERENCES
        Admin(Admin_ID)
);

CREATE TABLE Class_Location (
    Class_ID CHAR(5) NOT NULL,
    Class_Location VARCHAR2(20) NOT NULL,
    CONSTRAINT FK_ClassID FOREIGN KEY (Class_ID) REFERENCES
        Class(Class_ID),
    CONSTRAINT PK_Class_Location PRIMARY KEY (Class_ID, Class_Location)
);

CREATE TABLE Emergency_Contact (
    Phone_Number VARCHAR2(15) NOT NULL PRIMARY KEY,

```

```

        FName VARCHAR2(15) NOT NULL,
        LName VARCHAR2(15) NOT NULL,
        Address VARCHAR2(50) NOT NULL,
        Relation VARCHAR2(10) NOT NULL
    );

CREATE TABLE Student (
    Student_ID CHAR(5) NOT NULL PRIMARY KEY,
    FName VARCHAR2(15) NOT NULL,
    LName VARCHAR2(15) NOT NULL,
    Bdate DATE NOT NULL,
    Sex CHAR NOT NULL,
    Address VARCHAR2(50) NOT NULL,
    YearLevel INT NOT NULL,
    PhNum VARCHAR2(15) NOT NULL,
    Admin_ID CHAR(5) NOT NULL,
    Emergency_Number VARCHAR2(15) NOT NULL,
    CONSTRAINT FK_StudentAdmin FOREIGN KEY (Admin_ID) REFERENCES
        Admin(Admin_ID),
    CONSTRAINT FK_StudentContact FOREIGN KEY (Emergency_Number)
        REFERENCES Emergency_Contact(Phone_Number)
);

CREATE TABLE Takes (
    Student_ID CHAR(5) NOT NULL,
    Class_ID CHAR(5) NOT NULL,
    CONSTRAINT FK_StudentTakes FOREIGN KEY (Student_ID) REFERENCES
        Student(Student_ID),
    CONSTRAINT FK_ClassTakes FOREIGN KEY (Class_ID) REFERENCES
        Class(Class_ID),
    CONSTRAINT PK_Takes PRIMARY KEY (Student_ID, Class_ID)
);

CREATE TABLE Homework (
    Homework_ID CHAR(5) NOT NULL PRIMARY KEY,
    Date_Created DATE NOT NULL,
    Date_Due DATE NOT NULL,
    Grade VARCHAR2(15) NOT NULL
);

CREATE TABLE Student_Homework (
    Student_ID CHAR(5) NOT NULL,
    Homework_ID CHAR(5) NOT NULL,
    CONSTRAINT FK_StudentHW FOREIGN KEY (Student_ID) REFERENCES
        Student(Student_ID),
    CONSTRAINT FK_HWID FOREIGN KEY (Homework_ID) REFERENCES
        Homework(Homework_ID),
    CONSTRAINT PK_Student_Homework PRIMARY KEY (Student_ID,

```



```

        Homework_ID)
    );

CREATE TABLE Report (
    Report_ID CHAR(5) NOT NULL PRIMARY KEY,
    Report_Contents VARCHAR(150) NOT NULL,
    Date_Created DATE NOT NULL
);

CREATE TABLE Student_Report (
    Student_ID CHAR(5) NOT NULL,
    Report_ID CHAR(5) NOT NULL,
    CONSTRAINT FK_StudentReport FOREIGN KEY (Student_ID) REFERENCES
        Student(Student_ID),
    CONSTRAINT FK_ReportID FOREIGN KEY (Report_ID) REFERENCES
        Report(Report_ID),
    CONSTRAINT PK_Student_Report PRIMARY KEY (Student_ID, Report_ID)
);

INSERT INTO Department VALUES
('12345',TO_DATE('20-05-2011','DD-MM-YYYY'));
INSERT INTO Department VALUES
('54321',TO_DATE('10-04-1981','DD-MM-YYYY'));
INSERT INTO Department VALUES
('24680',TO_DATE('15-03-1991','DD-MM-YYYY'));
COMMIT;

INSERT INTO Department_Name VALUES('12345','English');
INSERT INTO Department_Name VALUES('54321','Maths');
INSERT INTO Department_Name VALUES('24680','Science');
COMMIT;

INSERT INTO Admin VALUES('59988','Oliver','Griffin','oliver.griffin@gmail.com');
INSERT INTO Admin VALUES('33333','James','Bond','jame_bond@yahoo.co.nz');
INSERT INTO Admin VALUES('34443','John','Power','john_power@hotmail.com');
COMMIT;

INSERT INTO Teacher VALUES('11115','George','Churton',
TO_DATE('20-05-2001','DD-MM-YYYY'),'M','16 George Street, Dunedin, New
Zealand','60000','12345','33333');
INSERT INTO Teacher VALUES('22225','Ben','Scobie',
TO_DATE('10-01-2001','DD-MM-YYYY'),'M','50 Dundas Street, Dunedin, New
Zealand','105000','54321','59988');
INSERT INTO Teacher VALUES('33335','Calvin','Pang',
TO_DATE('14-03-2000','DD-MM-YYYY'),'M','32 Queen Street, Auckland, New
Zealand','199999','24680','34443');
COMMIT;

```

```
INSERT INTO Department_Mgr VALUES ('12345','11115');
INSERT INTO Department_Mgr VALUES ('54321','22225');
INSERT INTO Department_Mgr VALUES ('24680','33335');
COMMIT;
```

```
INSERT INTO Teacher_PhoneNum VALUES('11115','02108307742');
INSERT INTO Teacher_PhoneNum VALUES('22225','0274647182');
INSERT INTO Teacher_PhoneNum VALUES('33335','0225556987');
COMMIT;
```

```
INSERT INTO Class VALUES('CS334','Computer Design',13,'24680','11115','59988');
INSERT INTO Class VALUES('MT205','Calculus',12,'54321','33335','34443');
INSERT INTO Class VALUES('ENG11','Reading',11,'12345','22225','33333');
COMMIT;
```

```
INSERT INTO Class_Location VALUES('CS334','Tech Lab 2');
INSERT INTO Class_Location VALUES('MT205','Room 6');
INSERT INTO Class_Location VALUES('ENG11','Room 7');
COMMIT;
```

```
INSERT INTO Emergency_Contact VALUES('0210220498','John','Dudson','25 George St,
Dunedin, New Zealand','Father');
INSERT INTO Emergency_Contact VALUES('0347776182','Emma','Johnson','50 Dundas St,
Dunedin, New Zealand','Mother');
INSERT INTO Emergency_Contact VALUES('034777618','Sandy','Smith','16 Hume St,
Dunedin, New Zealand','Mother');
COMMIT;
```

```
INSERT INTO Student
VALUES('72829','Oliver','Smith',TO_DATE('04-04-2007','DD-MM-YYYY'),'M','50 Dundas st,
Dunedin, New Zealand','11','022839190','59988','034777618');
INSERT INTO Student
VALUES('33229','Calvin','Johnson',TO_DATE('02-01-2006','DD-MM-YYYY'),'M','16 Hume st,
Dunedin, New Zealand','12','022736182','34443','0210220498');
INSERT INTO Student
VALUES('88355','Candice','Dudson',TO_DATE('24-12-2005','DD-MM-YYYY'),'M','25 George
Street, Dunedin, New Zealand','13','022343211','33333','0210220498');
COMMIT;
```

```
INSERT INTO Takes VALUES('72829','CS334');
INSERT INTO Takes VALUES('33229','MT205');
INSERT INTO Takes VALUES('88355','ENG11');
COMMIT;
```

```
INSERT INTO Homework
VALUES('HW_A1',TO_DATE('02-04-2022','DD-MM-YYYY'),TO_DATE('12-04-2006','DD-MM-
YYYY'),'Excellence');
```

```
INSERT INTO Homework
VALUES('HW_A3',TO_DATE('14-05-2022','DD-MM-YYYY'),TO_DATE('19-05-2006','DD-MM-
YYYY'),'Not Achieved');
INSERT INTO Homework
VALUES('HW_A6',TO_DATE('25-05-2022','DD-MM-YYYY'),TO_DATE('02-06-2006','DD-MM-
YYYY'),'Achieved');
COMMIT;
```

```
INSERT INTO Student_Homework VALUES('72829','HW_A1');
INSERT INTO Student_Homework VALUES('33229','HW_A3');
INSERT INTO Student_Homework VALUES('88355','HW_A6');
COMMIT;
```

```
INSERT INTO Report VALUES('RP201','Student received an
A+',TO_DATE('15-05-2022','DD-MM-YYYY'));
INSERT INTO Report VALUES('RP102','Student received a
B+',TO_DATE('15-05-2022','DD-MM-YYYY'));
INSERT INTO Report VALUES('RP404','Student received a
C-',TO_DATE('15-05-2022','DD-MM-YYYY'));
COMMIT;
```

```
INSERT INTO Student_Report VALUES('72829','RP201');
INSERT INTO Student_Report VALUES('33229','RP102');
INSERT INTO Student_Report VALUES('88355','RP404');
COMMIT;
```

## **6. Teamwork Summary**

Through utilising Microsoft teams our group was able to effectively work through the tasks associated with this assignment.

We started by discussing the feedback we had received from assignment one, and collectively modified our ER-Diagram using apps.diagrams.net with Oliver leading this area. This meant we could all see changes to the diagram in real-time. The use of this and video call on teams made every member of the team accountable and a part of the revision of the ER-Diagram.

After this we followed the seven step algorithm to convert the basic ER diagram into a relational schema on apps.diagrams.net-- ensuring the contribution and discussion of all team members with Calvin leading this area.

After the relational mapping was achieved, we attempted normalisation step by step for each relation from 1NF to BCNF. Discussions around this were achieved through team meetings and this section was led by George and Calvin.

After normalisation had been finished from 1NF to BCNF, we re-drew the relational mapping so that it would be easier to visualise and complete the sql part of the assessment. We then discussed how the sql code would look like, which was led by Ben.

Overall we found this assignment challenging and very good learning as we were able to apply our skills to a real world example and see how database systems are used in day to day systems like a school.