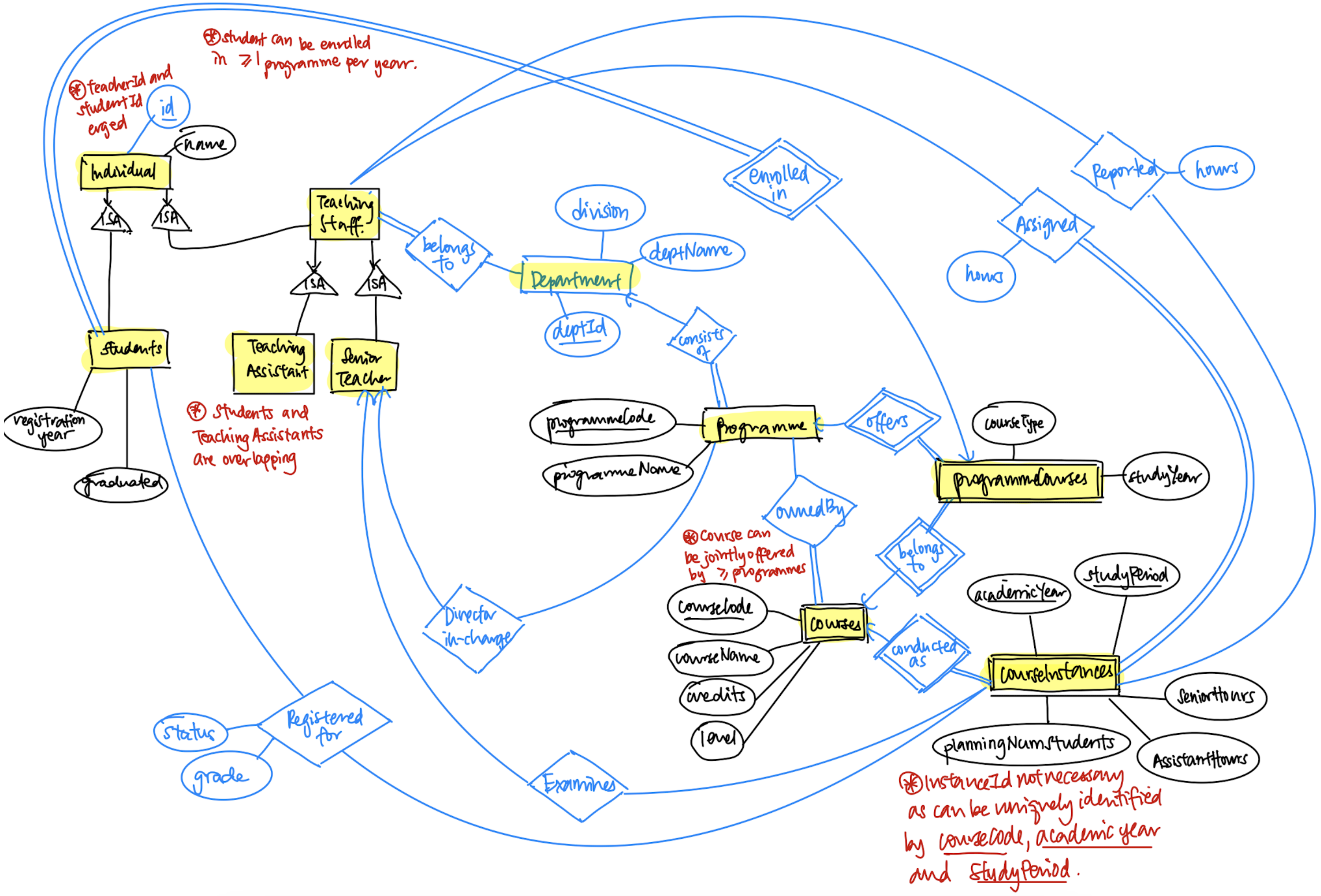
DIT930 Advanced Database - Assignment 1

Group 16: Yeoh Hui Qing, Erik Rosvall



⊛ teacherId and studentId erged

⊛ student can be enrolled in ≥ 1 programme per year.

⊛ Students and Teaching Assistants are overlapping

⊛ Course can be jointly offered by ≥ programmes

⊛ InstanceId not necessary as can be uniquely identified by courseCode, academic year and studyPeriod.

Decision	**Rationale**
Extract studentId/teacherId for Students/SeniorTeacher/ PhdStudent into id under Individual entity SeniorTeachers, PhdStudent and registered students are all classified under a TeachingStaff class entity.	Students who are also TeachingStaff would have both studentId and teacherId otherwise, which may not be necessary since they can be identified with a single id. Additionally from a school database system point of view it could be easier to query for individuals in the school from a single unique id than have separate portals to check for student teaching assistant records.
Created a Department entity that has the following attributes: deptId, deptName, division	A Department is clearly involved in the creation of Programmes and TeachingStaffs also belong to some Department, hence it made sense to create an entity for it rather than storing it as an attribute in separate entities.
director attribute of a Programme is extracted into a relationship between the SeniorTeacher and the Programme	Since the id of SeniorTeacher is a reference key for the director attribute of a Programme, it is easier to model this as a relationship between the two entities.
CourseInstance does not require an instanceId	(courseCode, studyPeriod, academicYear) will be able to uniquely identify the CourseInstance
examiner attribute of a CourseInstance is extracted into a relationship between the SeniorTeachers and the CourseInstance	Since the id of SeniorTeacher is a reference key for the examiner attribute of a CourseInstance, it is easier to model this as a relationship between the two entities.
CoursePlanning only need the attributes planningNumStudent, assistantHours, seniorHours	CoursePlanning does not need the course attribute since it gets the courseCode from the CourseInstance (which gets it from the PlanningCourse, etc).
AssignedHours and ReportedHours are represented as relationships between the TeachingStaff and CourseInstance than as independent entities	AssignedHours and ReportedHours does not make much sense to exist on their own, and since they both have attributes that are reference keys to CourseInstances, they can be modelled as attributes attached to the Assigned and Reported relationship instead. Reported relationship from CourseInstance can be 0..N as there could be no one reporting to the course, but for Assigned relationship it should be 1..N since there should be at least one teacher conducting the course instance for it to exist.
Registrations are modelled as a relationship between Students and contains the attributes status and grade	We get the studentId from the theudent relations, and the CourseId we get from the CourseInstance.
Merge CoursePlanning and CourseInstanse to one entity	The attributes of CoursePlanning does not serve any real purpose on its own, and we are also unable to uniquely identify CoursePlanning since it does not have an appropriate primary key, therefore it is more efficient to combine the two entities to one.

Additional remarks regarding Student:

- Students could be a part-time employee (teacher assistant) or a Phd student, however the Phd student is a full-time employee.
- Students can register to an instance of a course, however this is not a requirement. A student can be in the system without taking any courses.
- A student can be a teacher assistant, however it is not a must. We can distinguish a teacher assistant by id (since they are overlapping), the teacher assistant has the same id as the student id (inherit from the individual entity).