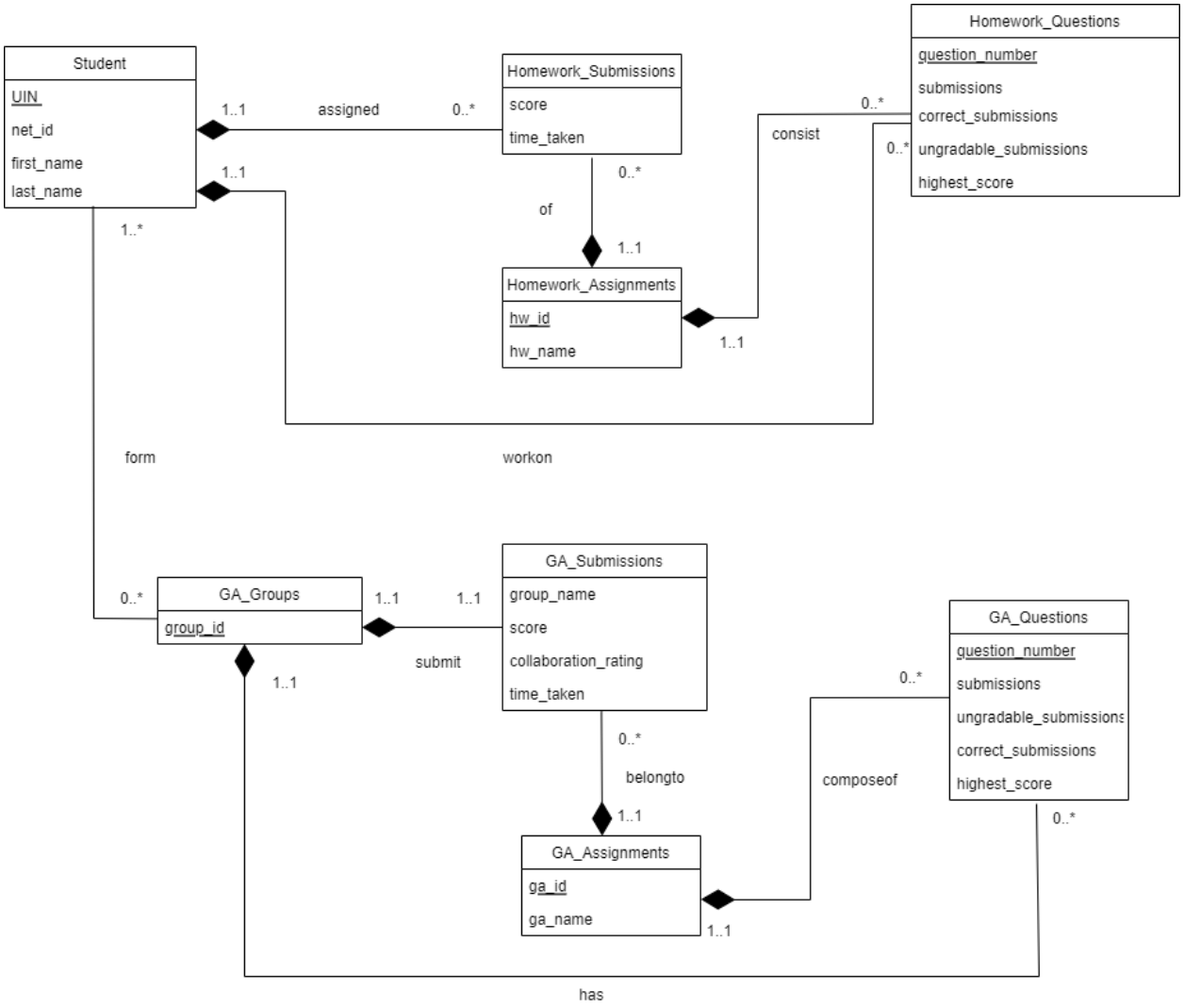
UML Diagram


Student
UIN
net_id
first_name
last_name
1..1
assigned
0..*
Homework_Submissions
score
time_taken
0..*
of
1..1
Homework_Assignments
hw_id
hw_name
1..1
0..*
consist
0..*
Homework_Questions
question_number
submissions
correct_submissions
ungradable_submissions
highest_score
1..1
1..*
form
workon
0..*
GA_Groups
group_id
1..1
1..1
submit
GA_Submissions
group_name
score
collaboration_rating
time_taken
1..1
0..*
belongto
1..1
GA_Assignments
ga_id
ga_name
1..1
0..*
composeof
GA_Questions
question_number
submissions
ungradable_submissions
correct_submissions
highest_score
0..*
has

Explanations of Entity Sets and Relationships

Student: has student information

Homework_Submissions: has homework submission statistics, is the **final** homework submission a student attempts
Homework_Assignments: has name of homework assignments
Homework_Questions: has question submission statistics for a particular (final) homework submission

GA_Groups: has GA group information to contain what students are in a specific group
GA_Submissions: has GA submission statistics, is the **final** homework submission a student attempts
GA_Assignments: has name of GA assignments
GA_Questions: has question submission statistics for a particular (final) GA submission

- "Submissions" here refers to the number of times a student submits an individual question

Each student can submit multiple homeworks, but each submission belongs to only one student
Each homework assignment can have multiple submissions (among multiple students), but each homework submission belongs to only one assignment
Each homework question is done by only one student, but each student can do multiple homework questions
Each homework question belongs to only one homework assignment, but each homework assignment can have multiple questions

Each student can be in multiple GA groups, and each GA group has at least one student
[Repeat homework relationships here, but now for groups]

Relationship Schema:

Student(UIN: INT [PK], net_id VARCHAR(10), first_name VARCHAR(50), last_name VARCHAR(50))

Homework_Assignments(hw_id: INT [PK], hw_name: VARCHAR(50))

Homework_Questions(hw_id: INT [PK][FK to Homework_Assignments.hw_id], question_number: INT [PK], UIN: INT [PK][FK to Student.UIN], submissions: INT, correct_submissions INT, ungradable_submissions INT, highest_score: REAL)

Homework_Submissions(UIN: INT [PK][FK to Student.UIN], hw_id: INT [PK][FK to Homework_Assignment.hw_id], score: REAL, time_taken: REAL)

GA_Groups(group_id: INT [PK])

GA_Submissions(group_id: INT [PK][FK to GA_Groups.group_id], ga_id: INT [PK][FK to GA_Assignments.ga_id], group_name: VARCHAR(20), score INT, collaboration_rating: INT, time_taken: REAL)

GA_Assignments(ga_id: INT [PK], ga_name: VARCHAR[20])

GA_Questions(ga_id: INT [PK][FK to GA_Assignment.ga_id], group_id: INT [PK][FK to GA_Groups.group_id], question_number: INT [PK], submissions: INT, ungradable_submissions: INT, correct_submissions: INT, highest_score: REAL)

SQL DDL Commands:

```
CREATE TABLE Students (
      uin INT PRIMARY KEY,
      net_id VARCHAR(10),
      first_name VARCHAR(50),
      last_name VARCHAR(50)
);

CREATE TABLE Homework_Assignments (
      hw_id INT PRIMARY KEY,
      score REAL,
);



CREATE TABLE Homework_Questions (
      hw_id INT FOREIGN KEY REFERENCES Homework_Assignments(hw_id),
      uin INT FOREIGN KEY REFERENCES Students(uin),
      question_number INT,
      submissions INT,
      correct_submissions INT,
      ungradable_submissions INT,
      highest_score REAL,
      PRIMARY KEY(question_number, hw_id, uin)
);



CREATE TABLE Homework_Submissions (
      hw_id INT FOREIGN KEY REFERENCES Homework_Assignments(hw_id),
      uin INT FOREIGN KEY REFERENCES Students(uin),
      Score REAL,
      Time_taken REAL,
      PRIMARY KEY(hw_id, UIN)
);

CREATE TABLE GA_Groups (
      group_id INT PRIMARY KEY,
);


CREATE TABLE GA_Submissions (
      group_id INT FOREIGN KEY REFERENCES GA_Groups(group_id),
      ga_id INT FOREIGN KEY REFERENCES GA_Assignments(ga_id),
```

```
      group_name VARCHAR(20),
      score INT,
      Collaboration_rating INT,
      Time_taken REAL
      PRIMARY KEY (group_id, ga_id)
);

CREATE TABLE GA_Assignments (
      ga_id INT PRIMARY KEY,
      ga_name VARCHAR[20]
);


CREATE TABLE GA_Questions (
      group_id INT FOREIGN KEY REFERENCES GA_Groups(group_id),
      ga_id INT FOREIGN KEY REFERENCES GA_Assignments(ga_id),
      question_number INT,
      submissions INT,
      ungradable_submissions INT,
      correct_submissions INT,
      highest_score REAL,
      PRIMARY KEY(question_number, group_id, ga_id)
);
```