Jacob Karcz
CS340: Databases
Final Project

Athletes Database

Database Outline Summary

This is a database linking athletes, to a diet, a focus, and a list of sports they participate in. This is something that might be used by personal trainers or coaches to track their athletes. Of course more tables would be beneficial in that scenario to track progress, so perhaps it's a nutritionist or a life coach's database. This is a convoluted field with many disciplines, each discipline of course has its own core values and they will differ depending on who you ask. Being such a convoluted area, I think this simplified outlook on the relationships can be beneficial.

Essentially, an athlete should have a focus; this is the goal that they are working towards. Whether it's completing an iron man, losing weight, gaining muscle, all of these can be defined by a focus. Each focus has sports and/or activities that will help the athlete achieve that goal. Furthermore, diets can be tailored specifically for a focus as well. Even if two athletes are running, if one is running to lose weight and the other is running to complete a marathon, they will have very different dietary needs. However, as an athlete progresses, their goals might change and their focus and diet may change accordingly as well.

For example, if an athlete wants to complete an iron man, their focus will be cardio (or an extreme version of it if it exists), their diet will be a high calorie diet that will energize them for training and provide protein for quick recovery, the athlete will participate in long distance running, long distance road cycling, swimming (pool), swimming (open water), and if they have time I'm sure they also enjoy hiking and some other stuff. Perhaps another athlete is looking to lose weight, so they have the appropriate diet, focus, and sports as well.

To simplify things on the coach's side, sports and diets are linked by a focus. Diets should be geared at a specific focus so it is no surprise that a diet has a focus. Sports have focuses as well, if someone is trying to bulk up they will want to focus on lifting weights, not running marathons so the coach can suggest a list of diets and/or sports based on their focus.

Jacob Karcz
CS340: Databases
Final Project

Database Outline

This database is for those that track athletes to better help them achieve their goals.

An athlete's goal is represented by a focus, this can be cardio, weight loss, bodybuilding, physical therapy (recovery), etc.

Athletes must have an ID, a first name, a last name, a focus, and a diet. They may also have a date of birth, gender, fitness level, and interests. Fitness level is a number from 1 – 10 to categorize the athlete's current level of fitness, and the athlete's interests is a short description of their goals. Athletes have one focus at a time, and they can only have one diet at a time to help them achieve their goal. They must have a diet and a focus. They also participate in at least one sport (otherwise they wouldn't be athletes). The athlete may change over time so they may update or correct any of the fields they are linked to, except sports because an athlete never quits.

Sports have an ID, a unique name, and at least one focus.

Foci have an ID and a type, they may or may not have sports and or diets associated with them. Each focus type is unique.

Diets have an ID, a name, a type, a focus, and (optionally) a description. The name identifies the diet, so it must be unique; the type describes the genre of diet (ie: high protein, paleo, glutten-free, low-fat, vegan); the focus is the focus that it caters to (ie high protein caters to bodybuilding); and the description is a short description of the diet.

Since diets and sports are potentially linked by a common focus, it is possible to view sports and diets filtered by a focus.

Athletes may be created, updated, and deleted using the forms provided. Updates and deletions are permanent.

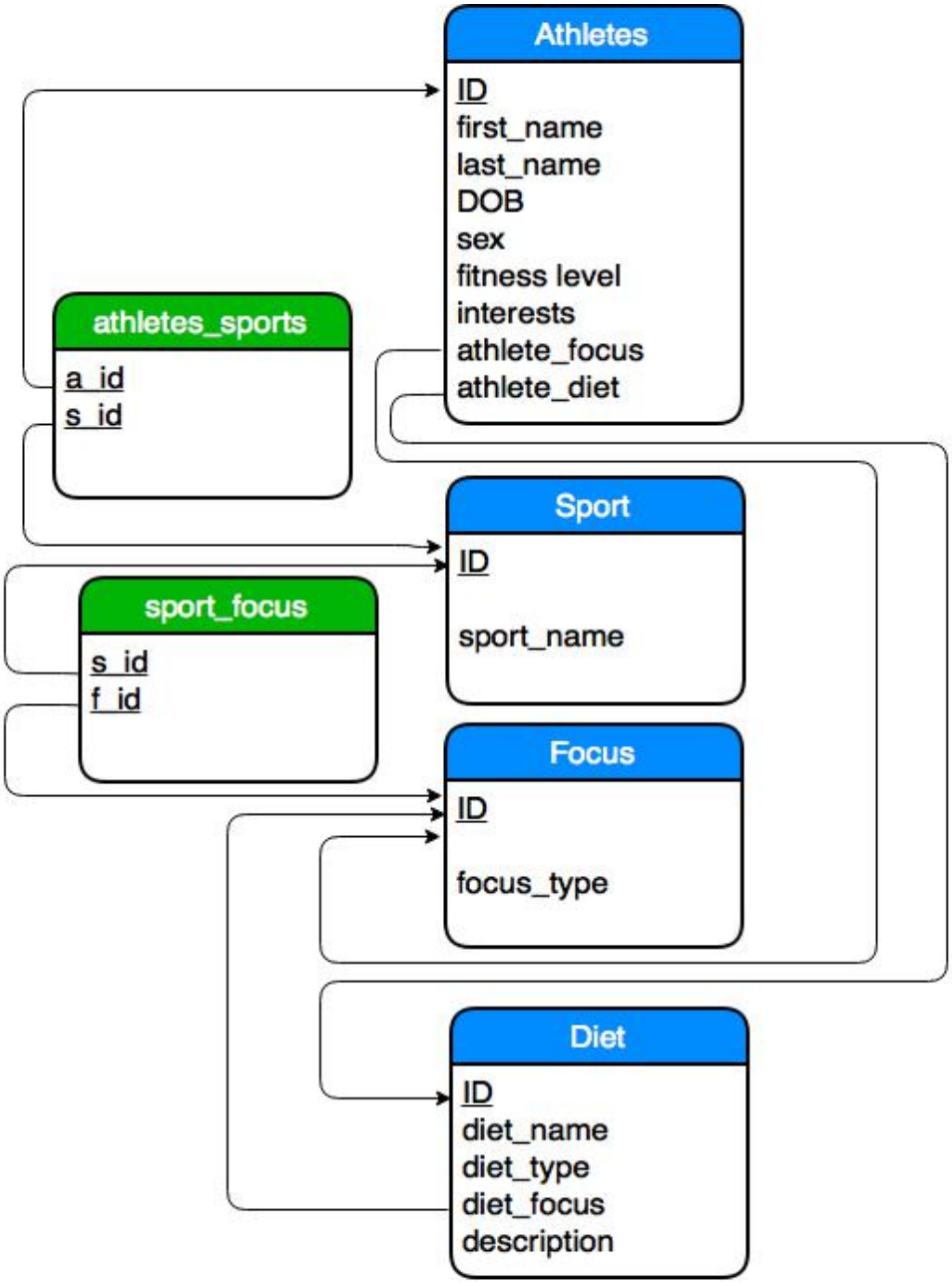
Sports, diets, and foci cannot be erased because that would potentially affect the other entities; only athletes may be erased because their deletion will not have a negative impact on the database.

All entities may be updated, however, as the updates can cascade through to any entities linked with the update. To minimize user issues, only athletes and diets may be updated through the website, please email the database administrator if you wish to update a sport, focus, or relationship between them and your request will be taken into consideration.

*Note the website cannot enforce the athletes-sports or focus-sport (M:M) relationships since the ID of the new entity is unknown at the time of creation. Thus sports must be added to the athlete separately after creating the athlete entity and a sport-focus relationship can only be created once both the sport and the focus exist.

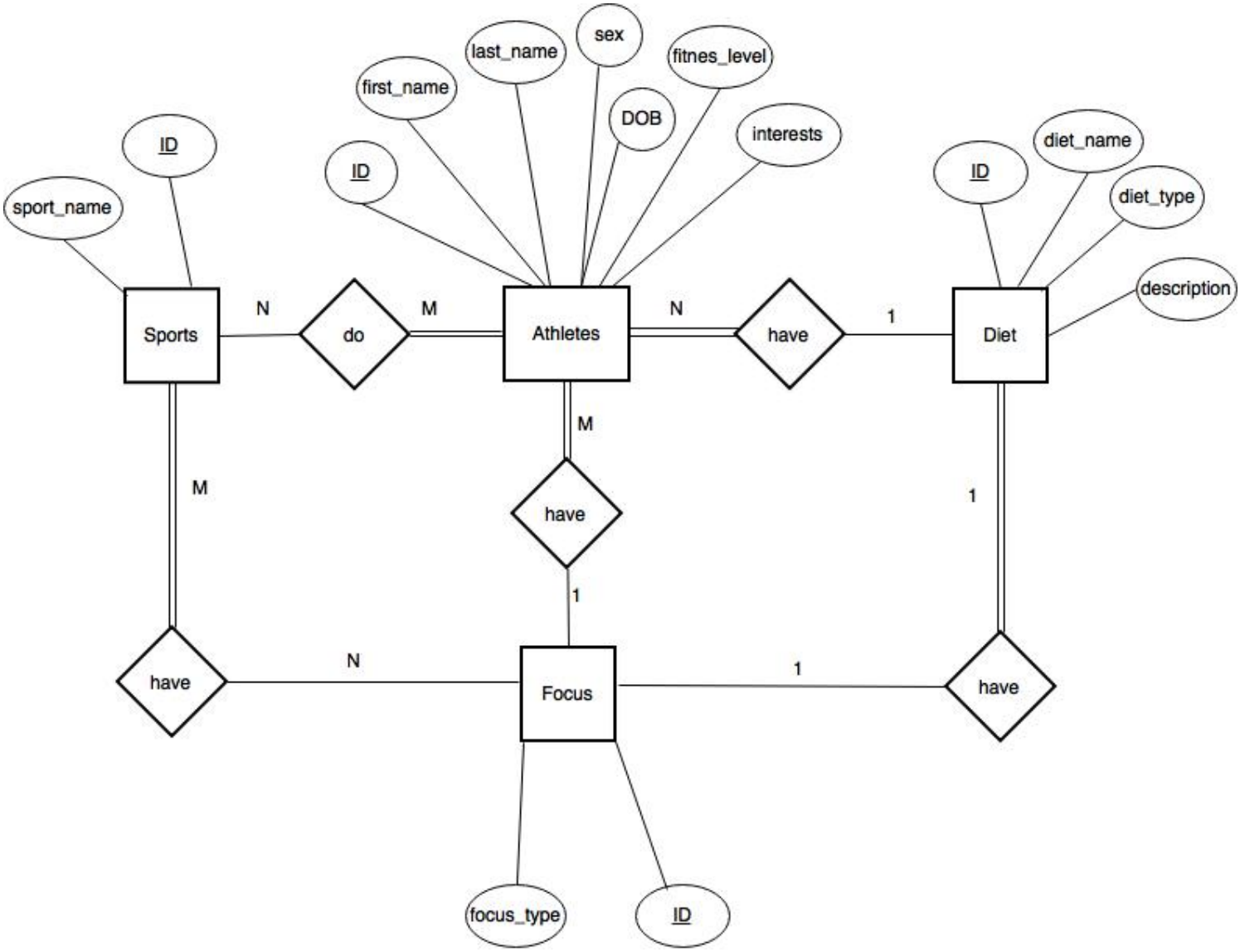
Jacob Karcz
CS340: Databases
Final Project

Database Schema



Jacob Karcz
CS340: Databases
Final Project

Database ER Diagram



Jacob Karcz
CS340: Databases
Final Project

TABLE CREATION QUERIES

```
DROP TABLE IF EXISTS `athletes_sports`;
DROP TABLE IF EXISTS `sport_focus`;
DROP TABLE IF EXISTS `sport`;
DROP TABLE IF EXISTS `athletes`;
DROP TABLE IF EXISTS `diet`;
DROP TABLE IF EXISTS `focus`;


CREATE TABLE focus (
	id INT NOT NULL AUTO_INCREMENT,
	focus_type VARCHAR(255) NOT NULL,
	PRIMARY KEY (id),
	UNIQUE KEY (focus_type)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE sport (
	id INT NOT NULL AUTO_INCREMENT,
	sport_name VARCHAR(255) NOT NULL,
	-- sport_type ?????????,
	PRIMARY KEY (id),
	UNIQUE KEY (sport_name)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE diet (
	id INT NOT NULL AUTO_INCREMENT,
	diet_name VARCHAR(255) NOT NULL,
	diet_type VARCHAR(255) NOT NULL,
	diet_focus INT NOT NULL,
	description TEXT,
	PRIMARY KEY (id),
	UNIQUE KEY (diet_name),
	FOREIGN KEY (diet_focus) REFERENCES focus (id)
		ON UPDATE CASCADE
		ON DELETE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE athletes (
	id INT NOT NULL AUTO_INCREMENT,
	first_name VARCHAR(255) NOT NULL,
	last_name VARCHAR(255) NOT NULL,
```

```
	DOB DATE,
	sex VARCHAR(10),
	fitness_level INT UNSIGNED,
	athlete_focus INT NOT NULL,
	athlete_diet INT NOT NULL,
	interests TEXT,
	PRIMARY KEY (id),
	CONSTRAINT ath_foc FOREIGN KEY (athlete_focus) REFERENCES focus (id)
		ON UPDATE CASCADE
		ON DELETE NO ACTION,
	FOREIGN KEY (athlete_diet) REFERENCES diet (id)
		ON UPDATE CASCADE
		ON DELETE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE athletes_sports (
	a_id INT NOT NULL,
	s_id INT NOT NULL,
	-- start_date DATE NOT NULL,
	-- intensity INT, -- ?
	PRIMARY KEY (a_id, s_id),
	KEY (a_id),
	CONSTRAINT fk_ath FOREIGN KEY (a_id) REFERENCES athletes (id)
		ON UPDATE CASCADE
		ON DELETE CASCADE,
	CONSTRAINT fk_sport FOREIGN KEY (s_id) REFERENCES sport (id)
		ON UPDATE CASCADE
		ON DELETE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE sport_focus (
	f_id INT NOT NULL,
	s_id INT NOT NULL,
	-- skill_level INT NOT NULL, -- ?
	PRIMARY KEY ( f_id, s_id),
	KEY (f_id),
	CONSTRAINT fk_sprt FOREIGN KEY (s_id) REFERENCES sport (id)
		ON UPDATE CASCADE
		ON DELETE NO ACTION,
	CONSTRAINT fk_foc FOREIGN KEY (f_id) REFERENCES focus (id)
		ON UPDATE CASCADE
		ON DELETE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Jacob Karcz
CS340: Databases
Final Project

```
-- SEED TABLES --

--  seed focus
LOCK TABLES focus WRITE;
/*!40000 ALTER TABLE `focus` DISABLE KEYS */;
INSERT INTO focus VALUES (1, 'bodybuilding'), (2, 'cardio'), (3, 'strength'), (4, 'flexibility'),
                         (5, 'agility'), (6, 'weight loss'), (7, 'recovery'),  (8, 'mindfulness');
/*!40000 ALTER TABLE `focus` ENABLE KEYS */;
UNLOCK TABLES;

-- seed diets
LOCK TABLES diet WRITE;
/*!40000 ALTER TABLE `diet` DISABLE KEYS */;
INSERT INTO diet VALUES
        (1, 'SoCal', 'low calorie', 6, 'not the best option'),
        (2, 'shredded', 'low fat', 1, 'a low fat diet'),
        (3, 'paleo', 'low carb', 6, 'a high risk high result short term diet'),
        (4, 'zen', 'balanced', 8, 'a balanced diet, ideal for maintaining a healthy lifestyle'),
        (5, 'vegan delight', 'balanced – high protein', 4, 'a high protein plant based diet'),
        (6, 'Beastly', 'high protein', 3, 'a high protein diet for building big muscles'),
        (7, 'finish line', 'high carb', 2, 'provides abundant energy for race day'),
        (8, 'fuel up', 'high calorie', 5, 'energy for cardio + strength');
/*!40000 ALTER TABLE `diet` ENABLE KEYS */;
UNLOCK TABLES;

-- seed sports
LOCK TABLES sport WRITE;
/*!40000 ALTER TABLE `sport` DISABLE KEYS */;
INSERT INTO sport VALUES (1, 'swimming'), (2, 'cycling'), (3, 'running'), (4, 'resistance training'),
                         (5, 'HIIT'), (6, 'Yoga'), (7, 'basketball'),  (8, 'touch football'),
                         (9, 'rugby'), (10, 'kayaking'), (11, 'soccer'), (12, 'ski'),
                         (13, 'snowboarding');
/*!40000 ALTER TABLE `sport` ENABLE KEYS */;
UNLOCK TABLES;

-- seed athletes
LOCK TABLES athletes WRITE;
/*!40000 ALTER TABLE `athletes` DISABLE KEYS */;
INSERT INTO athletes VALUES
(1, 'Jake', 'Klein', '1985-11-18', 'M', 7, 2, 4, 'triathlons, also yoga and strength training for balance'),
(2, 'Hillary', 'Jaimes', '1987-10-26', 'F', 8, 3, 6, 'bulking up for rock climbing');
/*!40000 ALTER TABLE `athletes` ENABLE KEYS */;
UNLOCK TABLES;
```

```
-- seed sport-focus relationship
-- INSERT INTO sport_focus -- (focus, sport)
LOCK TABLES sport_focus WRITE;
/*!40000 ALTER TABLE `sport_focus` DISABLE KEYS */;
INSERT INTO sport_focus VALUES
(2, 1), (6, 1), (7, 1), (3, 1), (2, 2), (2, 3), (6, 3), (5, 3), (3, 4), (1, 4), (6, 4), (7, 4), (1, 5), (6, 5), (5, 5),
(3, 5), (2, 5), (4, 6), (8, 6), (7, 6), (2, 7), (5, 7), (6, 7), (7, 7), (2, 8), (5, 8), (6, 8), (7, 8), (2, 9), (5, 9),
(6, 9), (2, 10), (3, 10), (6, 10), (7, 10), (2, 11), (5, 11), (6, 11), (6, 12), (7, 12), (6, 13), (7, 13);
/*!40000 ALTER TABLE `sport_focus` ENABLE KEYS */;
UNLOCK TABLES;

-- seed athlete-sport relationship
-- INSERT INTO athletes_sports -- (athlete, sport)
LOCK TABLES athletes_sports WRITE;
/*!40000 ALTER TABLE `athletes_sports` DISABLE KEYS */;
INSERT INTO athletes_sports VALUES
                    (1, 2), (1, 3), (1, 4), (1, 6), (1, 5), (1, 10), (2, 4), (2, 1), (2, 5), (2, 6), (2, 13);
/*!40000 ALTER TABLE `athletes_sports` ENABLE KEYS */;
UNLOCK TABLES;


-- Test Query: find sport and sport focus info for athlete id number 1
SELECT CONCAT(a.last_name, ', ', a.first_name) AS athlete,
                    s.sport_name AS sport, f.focus_type AS sportFocus FROM athletes a
INNER JOIN athletes_sports a_s ON a_s.a_id = a.id
INNER JOIN sport s ON s.id = a_s.s_id
INNER JOIN sport_focus sf ON sf.s_id = s.id
INNER JOIN focus f ON sf.f_id = f.id
WHERE a.id = 1;

-- Test Query: get all athlete info
SELECT a.first_name, a.last_name, a.DOB, a.sex, a.fitness_level,
          f.focus_type, d.diet_name, d.diet_type, s.sport_name
FROM athletes a
INNER JOIN focus f ON a.athlete_focus = f.id
INNER JOIN diet d ON a.athlete_diet = d.id
INNER JOIN athletes_sports a_s ON a_s.a_id = a.id
INNER JOIN sport s ON s.id = a_s.s_id;
```

Jacob Karcz
CS340: Databases
Final Project

GENERAL QUERIES

A. General SELECT queries to populate tables

1. Populate the athlete's table:

```
SELECT a. id, a.first_name, a.last_name, a.DOB, a.sex, a.fitness_level,
            f.focus_type, d.diet_name, d.diet_type, a.interests FROM athletes a
INNER JOIN focus f ON a.athlete_focus = f.id
INNER JOIN diet d ON a.athlete_diet = d.id
```

2. Populate the diets table:

```
SELECT d.id, d.diet_name, d.diet_type, f.focus_type, d.description FROM diet d
INNER JOIN focus f ON d.diet_focus = f.id
```

3. Populate the sports table:

```
SELECT *  FROM sports
```

4. Populate the focus table:

```
SELECT *  FROM focus
```

B. User input SELECT queries

1. View diets and sports filtered by focus*:

    i. Select from sports:
    ```
    SELECT s.sport_name FROM sport_focus sf
    INNER JOIN sport s ON s.id = sf.s_id
    WHERE sf.f_id = [focus ID]
    ```

    ii. Select from diets:
    ```
    SELECT diet_name, diet_type, description FROM diet
    WHERE diet_focus =  [focus ID]
    ```

2. View an athlete and their list of sports*:

    i. Select the athlete:
    ```
    SELECT a. id, a.first_name, a.last_name, a.DOB, a.sex, a.fitness_level,
                f.focus_type, d.diet_name, d.diet_type, a.interests FROM athletes a
    INNER JOIN focus f ON a.athlete_focus = f.id
    INNER JOIN diet d ON a.athlete_diet = d.id
    WHERE a.id = [athlete's ID]
    ```

    ii. Select the athlete's sports:
    ```
    SELECT s.sport_name FROM athletes_sports a_s
    INNER JOIN sport s ON s.id = a_s.s_id
    WHERE a_s.a_id = [athlete's ID]
    ```

* This functionality requires two queries; otherwise attributes will be removed or replicated depending on the syntax of the singular search.

C. INSERT queries

1. Add an athlete:

    ```
    INSERT INTO athletes (first_name, last_name, DOB, sex, fitness_level,
                athlete_focus, athlete_diet, interests)
    VALUES ( [ first name], [last name], [date of birth], [gender],
            [fitness level], [focus], [diet], [interests] )
    ```

2. Add a sport to an athlete's profile:

    ```
    INSERT INTO athletes_sports (a_id, s_id) VALUES ( [athlete ID], [sport ID] )
    ```

3. Add a sport:

    ```
    INSERT INTO focus (focus_type) VALUES ( [sport name] )
    ```

4. Add a focus:

    ```
    INSERT INTO sport (sport_name) VALUES ( [focus type] )
    ```

5. Add a sport to a focus (or a focus to a sport, depending on perspective)

   INSERT INTO sport_focus (f_id, s_id) VALUES ([focus ID], [sport ID])

6. Add a diet:

   INSERT INTO diet (diet_name, diet_type, diet_focus, description)
   VALUES ([diet name], [diet type], [focus ID], [diet description])

D. UPDATE queries

Note: the attributes may or may not be updated depending on the value of the website form's input field. If the field is blank, then the attribute will not be updated and the current value will be kept.

1. Update an athlete:

   UPDATE athletes SET first_name = [current first name, or updated attribute],
   last_name = [current last name, or updated attribute],
   DOB = [current date of birth, or updated attribute],
   sex = [current gender, or updated attribute],
   fitness_level = [current fitness level, or updated attribute],
   athlete_focus = [current focus, or updated attribute],
   athlete_diet = [current diet, or updated attribute],
   interests = [current interests, or updated attribute]
   WHERE id = [athlete ID]

2. Update a diet:

   UPDATE diet SET diet_name = [current diet name, or updated attribute],
   diet_type = [current diet type, or updated attribute],
   diet_focus = [current focus (ID), or updated attribute],
   description = [current diet description, or updated attribute]
   WHERE id = [diet ID]

E. DELETE queries

1. Delete an athlete:

   DELETE FROM athletes WHERE id = [athlete ID]

F. Queries to populate dropdown menus to select an entity ID

1. Display athlete names (but select value is the athlete ID):

   ```
   SELECT id, first_name, last_name FROM athletes
   ```

2. Display focus types or sport names (but select value is the entity ID):

   ```
   SELECT * FROM focus
   SELECT * FROM focus
   ```

3. Display diet information (but select value is the entity ID):

   ```
   SELECT d.id, d.diet_name, d.diet_type, f.focus_type, d.description FROM diet d
   INNER JOIN focus f ON d.diet_focus = f.id
   ```