Steven Ha
12/06/15
CS340
Final Project

OUTLINE

Over the past few years, mainstream media has been covering more climbing related news. Especially with Alex Honnald free soloing multi pitch routes and Tommy Caldwell and Kevin Jorgeson freeing the Dawn Wall project in Yosemite Valley. This media coverage is increasing the number of climbers in the US. Most start out in gyms and at some point move their climbing to the outdoors. Climbing is mainly broken down into two categories of importance. Bouldering which covers short powerful problems with the protection of bouldering pads. Sport climbing consists of climbing with the aid of a rope since the problems are taller and involve more endurance. This database will track the boulder problems and sport routes around the US, as well as monitoring the boulder and/or sport problems that climbers have completed. The database will help people find boulders or sport routes by location or grade. This can also be used to find the strongest climbers in the US and the hardest problems in the US.

DATABASE OUTLINE IN WORDS

The database is comprised of 10 tables total. 2 of those tables are static and are used to help rank the boulder and sport problems. The tables are prepopulated with an index for each known boulder and sport grade. These tables assist in ordering the boulder and sport problems using the indexes in these tables.

Boulders and Sport problems must have a name and a grade. Boulder and sport problems can have the same name and grade, but a boulder problem or sport problem can only exist once in a given location. Boulder and sport problems are kept track in their respective tables (boulder and sport). Specific boulder and sport problems need to exist in a certain location. The boulder_location and sport_location tables keep track of the locations of the problems. The rows in the table will contain the id of the boulder/sport id along with the id of the location. The locations table will be used to keep track of the name, city, and state where the climbing location exist. A climbing location is unique, duplicate locations cannot exist.

The climber table keeps track of the climbers. Climbers must have a first name, last name, and birthday. Climbers are unique, so no other climber will have the same name or birthday. The boulder_climber table will keep track of the climbers that have climbed a specific boulder. A climber can only complete a boulder problem once. The sport_climber table will keep track of the climbers that have climbed a specific sport problem. A climber can only complete a sport problem once. Boulder and sport problems can be climbed by multiple climbers.

TABLE CREATION QUERIES

```
-- table contains climber info--
CREATE TABLE `climber` (
	`id` int(11) NOT NULL AUTO_INCREMENT,
	`firstName` varchar(255) NOT NULL,
	`lastName` varchar(255) NOT NULL,
	`birthday` date NOT NULL,
	primary key (`id`),
	constraint unique (`firstName`, `lastName`, `birthday`)
) ENGINE=InnoDB;

--table contains boulder problems--
CREATE TABLE `boulder` (
	`id` int(11) NOT NULL AUTO_INCREMENT,
	`name` varchar(255) NOT NULL,
	`grade` varchar(255) NOT NULL,
	primary key(`id`)
) ENGINE=InnoDB;

--table contains sport problems--
CREATE TABLE `sport` (
	`id` int(11) NOT NULL AUTO_INCREMENT,
	`name` varchar(255) NOT NULL,
	`grade` varchar(255) NOT NULL,
	primary key(`id`)
) ENGINE=InnoDB;

--table contains locations where boulder or sport problems are --
CREATE TABLE `location` (
	`id` int(11) NOT NULL AUTO_INCREMENT,
	`name` varchar(255) NOT NULL,
	`city` varchar(255) NOT NULL,
	`state` varchar(255) NOT NULL,
	primary key (`id`),
	constraint unique (`name`, `city`, `state`)
) ENGINE=InnoDB;

--table contains information of boulders climbed by climbers--
CREATE TABLE `boulder_climber` (
  `cid` int(11) NOT NULL DEFAULT '0',
  `bid` int(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`cid`,`bid`),
  KEY `bid` (`bid`),
  CONSTRAINT `bc_climber` FOREIGN KEY (`cid`) REFERENCES `climber` (`id`),
  CONSTRAINT `bc_boulder` FOREIGN KEY (`bid`) REFERENCES `boulder` (`id`)
) ENGINE=InnoDB;

--table contains information of sport problems climbed by climbers--
CREATE TABLE `sport_climber` (
  `cid` int(11) NOT NULL DEFAULT '0',
  `sid` int(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`cid`,`sid`),
  KEY `sid` (`sid`),
  CONSTRAINT `sc_climber` FOREIGN KEY (`cid`) REFERENCES `climber` (`id`),
  CONSTRAINT `sc_boulder` FOREIGN KEY (`sid`) REFERENCES `sport` (`id`)
) ENGINE=InnoDB;

--table contains location of boulders--
CREATE TABLE `boulder_location` (
  `bid` int(11) NOT NULL DEFAULT '0',
  `lid` int(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`bid`,`lid`),
  KEY `lid` (`lid`),
  CONSTRAINT `bl_boulder` FOREIGN KEY (`bid`) REFERENCES `boulder` (`id`),
  CONSTRAINT `bl_location` FOREIGN KEY (`lid`) REFERENCES `location` (`id`)
) ENGINE=InnoDB;

--table contains location of sport problems--
CREATE TABLE `sport_location` (
  `sid` int(11) NOT NULL DEFAULT '0',
  `lid` int(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`sid`,`lid`),
  KEY `lid` (`lid`),
  CONSTRAINT `sl_sport` FOREIGN KEY (`sid`) REFERENCES `sport` (`id`),
  CONSTRAINT `sl_location` FOREIGN KEY (`lid`) REFERENCES `location` (`id`)
) ENGINE=InnoDB;

--table that contains the index of boulder grades, used to show ranking--
CREATE TABLE `boulder_Index`(
  `grade` varchar(255) NOT NULL DEFAULT '0',
  `index` int(11) NOT NULL DEFAULT '0'
) ENGINE=InnoDB;

--table that contains the index of sport grades, used to show ranking--
CREATE TABLE `sport_Index`(
  `grade` varchar(255) NOT NULL DEFAULT '0',
  `index` int(11) NOT NULL DEFAULT '0'
) ENGINE=InnoDB;
```

GENERAL USE QUERIES

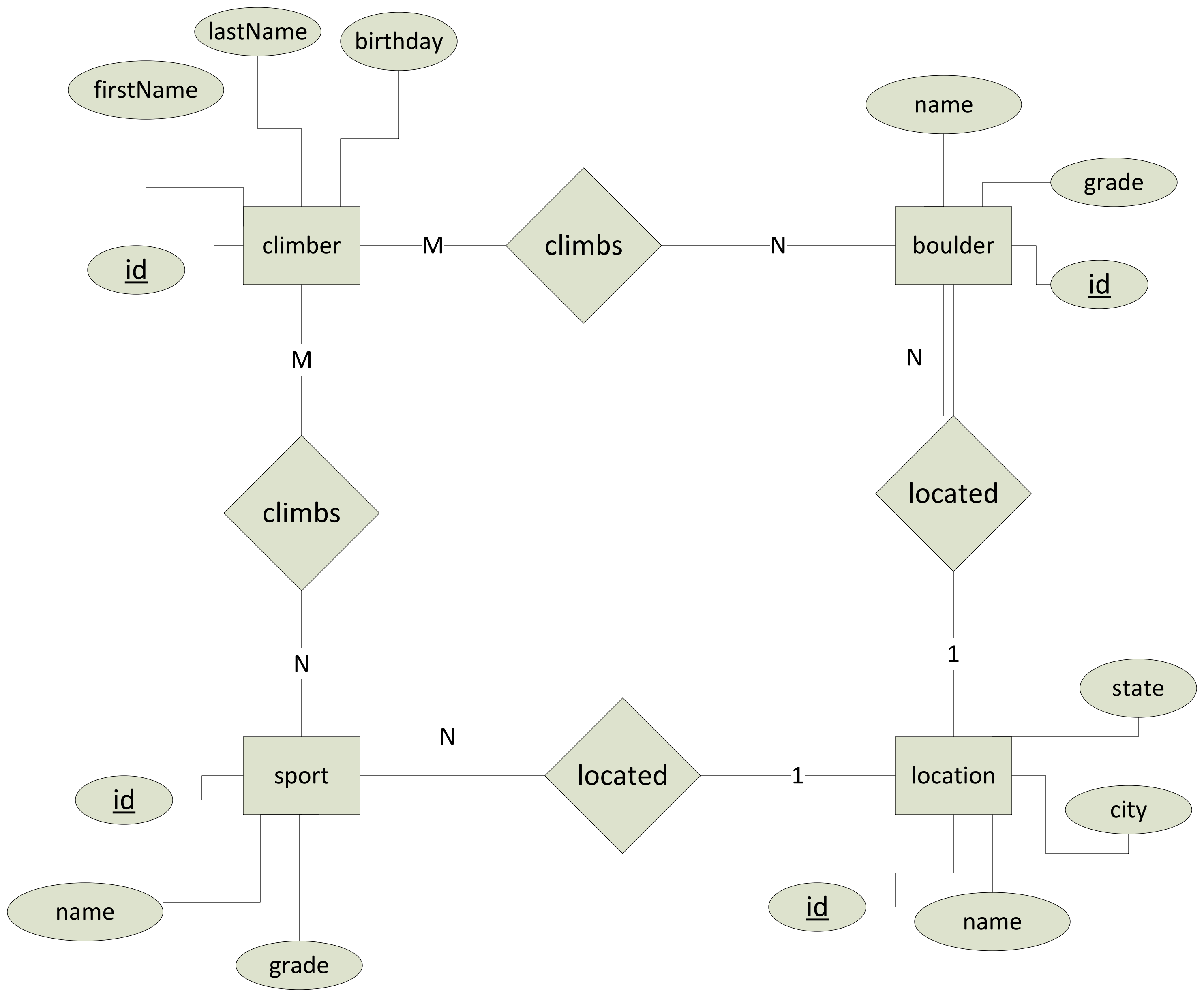
```
--fill boulder_Index table with prepopulated values--
INSERT INTO `boulder_Index` values ('V0',0),('V1',1),('V2',2),('V3',3),
('V4',4),('V5',5),('V6',6),('V7',7),('V8',8),('V9',9),('V10',10),('V11',11),
('V12',12),('V13',13),('V14',14),('V15',15);

--fill sport_Index table with prepopulated values--
INSERT INTO `sport_Index` values ('5.9',0),('5.10a',1),('5.10b',2),('5.10c',3),
('5.10d',4),('5.11a',5),('5.11b',6),('5.11c',7),('5.11d',8),('5.12a',9),
('5.12b',10),('5.12c',11),('5.12d',12),('5.13a',13),('5.13b',14),('5.13c',15),
('5.13d',16),('5.14a',17),('5.14b',18),('5.14c',19),('5.14d',20),('5.15a',21),
('5.15b',22),('5.15c',23);

--insert a climber into the climber database--
INSERT INTO `climber` (`firstName`, `lastName`, `birthday`) VALUES ([fName],[lName],[bDate]);

--insert a boulder into the boulder database--
INSERT INTO `boulder` (`name`, `grade`) VALUES ([bName],[bGrade]);

--insert a sport into the sport database--
INSERT INTO `sport` (`name`, `grade`) VALUES ([sName],[sGrade]);

--insert a sport into the sport database--
```

```
INSERT INTO `location` (`name`, `city`, `state`) VALUES ([locationName], [cityName],[stateName]);

--insert into the boulder climber database--
INSERT INTO `boulder_climber` (`cid`, `bid`) VALUES ([climber_id],[boulder_id]);

--insert into the sport climber database--
INSERT INTO `sport_climber` (`cid`, `sid`) VALUES ([climber_id],[sport_id]);

--insert into the sport location database--
INSERT INTO `sport_location` (`sid`, `lid`) VALUES ([sport_id],[location_id]);

--insert into the boulder location database--
INSERT INTO `boulder_location` (`did`, `lid`) VALUES ([boulder_id],[location_id]);

--show boulder info and location of boulder by user specified boulder grade--
SELECT b.name, b.grade, l.name FROM boulder b
INNER JOIN boulder_location bl ON bl.bid = b.id
INNER JOIN location l ON l.id = bl.lid
WHERE b.grade = [boulder_grade];

--show boulder info and location of boulder by user specified location--
SELECT b.name, b.grade, l.name FROM boulder b
INNER JOIN boulder_location bl ON bl.bid = b.id
INNER JOIN location l ON l.id = bl.lid
WHERE l.id = [location_id];

--show boulder info and location of boulder climbed by user specified climber--
SELECT b.name, b.grade, l.name FROM boulder b
INNER JOIN boulder_location bl ON bl.bid = b.id
INNER JOIN location l ON bl.lid = l.id
INNER JOIN boulder_climber bc ON bc.bid = b.id
WHERE bc.cid = [climber_id];

--show climber info and boulder info of climber that has climbed a user specified boulder grade--
SELECT c.firstName, c.lastName, c.birthday, b.name, b.grade, b.id, l.name FROM boulder_climber bc
INNER JOIN boulder b ON b.id = bc.bid
INNER JOIN climber c on bc.cid = c.id
INNER JOIN boulder_location bl ON bl.bid = b.id
INNER JOIN location l ON l.id = bl.lid
WHERE b.grade= [boulder_grade];

--show climber info and boulder info of climber that has climbed a user specified boulder problem--
SELECT c.firstName, c.lastName, c.birthday, b.name, b.id, b.grade, l.name FROM boulder_climber bc
INNER JOIN boulder b ON b.id = bc.bid
INNER JOIN climber c on bc.cid = c.id
INNER JOIN boulder_location bl ON bl.bid = b.id
INNER JOIN location l ON l.id = bl.lid
WHERE b.id= [boulder_id];
```

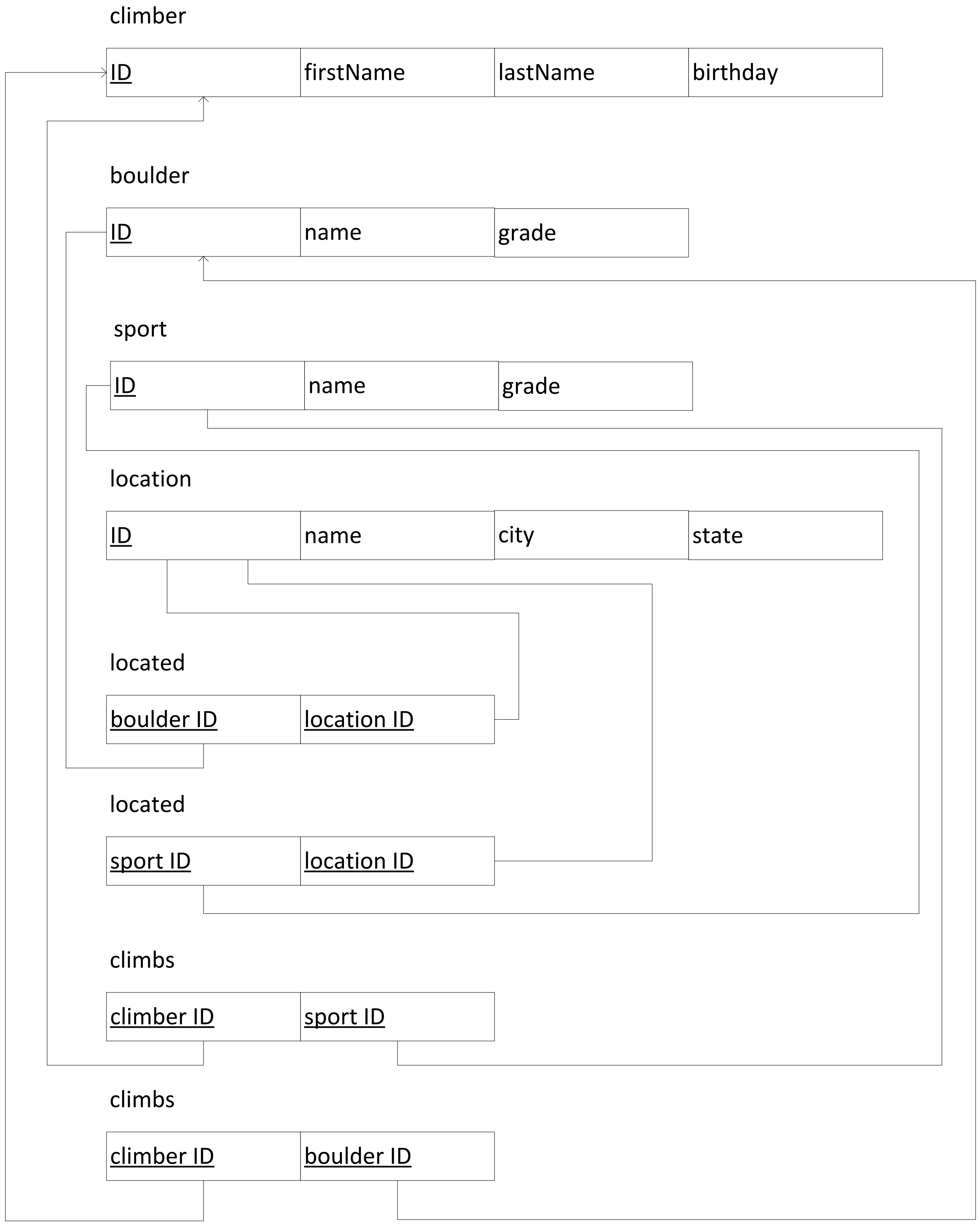
```
--show climber info and sport info of climber that has climbed a user specified sport grade--
SELECT c.firstName, c.lastName, c.birthday, s.name, s.grade, s.id, l.name FROM sport_climber sc
INNER JOIN sport s ON s.id = sc.sid
INNER JOIN climber c on sc.cid = c.id
INNER JOIN sport_location sl ON sl.sid = s.id
INNER JOIN location l ON l.id = sl.lid
WHERE s.grade= [sport_grade];

--show climber info and sport info of climber that has climbed a user specified sport problem--
SELECT c.firstName, c.lastName, c.birthday, s.name, s.id, s.grade, l.name FROM sport_climber sc
INNER JOIN sport s ON s.id = sc.sid
INNER JOIN climber c on sc.cid = c.id
INNER JOIN sport_location sl ON sl.sid = s.id
INNER JOIN location l ON l.id = sl.lid
WHERE s.id= [sport_id];

--show location info based on user specified state name--
SELECT l.name, l.city, l.state FROM location l
WHERE l.state = [state_name];

--show sport info and location of sport route for user specified sport grade--
SELECT s.name, s.grade, l.name FROM sport s
INNER JOIN sport_location sl ON sl.sid = s.id
INNER JOIN location l ON l.id = sl.lid
WHERE s.grade = [sport_grade];

--show sport info and location of sport route for user specified location--
SELECT s.name, s.grade, l.name FROM sport s
INNER JOIN sport_location sl ON sl.sid = s.id
INNER JOIN location l ON l.id = sl.lid
WHERE l.id = [location_id];

--show boulder info and location of boulder climbed by user specified climber--
SELECT s.name, s.grade, l.name FROM sport s
INNER JOIN sport_location sl ON sl.sid = s.id
INNER JOIN location l ON sl.lid = l.id
INNER JOIN sport_climber sc ON sc.sid = s.id
WHERE sc.cid = [climber_id];

--show all boulders along with their locations--
SELECT b.name, b.grade, l.name FROM boulder b
INNER JOIN boulder_location bl ON bl.bid = b.id
INNER JOIN location l ON l.id = bl.lid;

--show all climbers--
SELECT firstName, lastName, birthday FROM climber;

--show all sports along with their locations--
SELECT s.name, s.grade, l.name FROM sport s
```

```
INNER JOIN sport_location sl ON sl.sid = s.id
INNER JOIN location l ON l.id = sl.lid;

--delete boulder to climber relationship--
DELETE FROM `boulder_climber` WHERE cid=[climber_id] AND bid=[boulder_id];

--delete sport to climber relationship--
DELETE FROM `sport_climber` WHERE cid=[climber_id] AND sid=[sport_id];

--update boulder location--
UPDATE boulder_location SET lid=[location_id] WHERE bid=[boulder_id];

--update boulder problem--
UPDATE boulder SET name=[new_name], grade=[new_grade] WHERE id=[boulder_id];

--update climber--
UPDATE climber SET firstName= [new_firstName], lastName=[new_lastName], birthday=[new_birthday] WHERE id= [climber_id];

--update location--
UPDATE location SET name=[new_name], city=[new_city], state=[new_state] WHERE id=[location_id];

--update sport location--
UPDATE sport_location SET lid=[location_id] WHERE sid=[sport_id];

--update sport--
UPDATE sport SET name=[new_name], grade=[new_grade] WHERE id=[sport_id];
```

ER Diagram



Schema



climber

ID	firstName	lastName	birthday

boulder

ID	name	grade

sport

ID	name	grade

location

ID	name	city	state

located

boulder ID	location ID

located

sport ID	location ID

climbs

climber ID	sport ID

climbs

climber ID	boulder ID