```
--ASSIGNMENT 3
--JOHN HODSON
---------------------

--1
--SQL FOR QUESTION 1
---------------------
SELECT DISTINCT p.name
FROM CLIMBED c, PARTICIPATED p
WHERE c.PEAK = "Pilot Knob (S)" AND
        c.trip_id = p.TRIP_ID;
---------------------


--2
--SQL FOR QUESTION 2
---------------------
SELECT DISTINCT c.peak
FROM PARTICIPATED p, climbed c
WHERE p.name = "MARK" AND
     p.trip_id = c.trip_id;
---------------------


--3
--SQL FOR QUESTION 3
---------------------
SELECT DISTINCT p.name
FROM PEAK m, CLIMBED c, PARTICIPATED p
WHERE m.diff = 5 AND
        c.PEAK = m.name AND
     p.TRIP_ID = c.trip_id;
---------------------


--4
--SQL FOR QUESTION 4
---------------------
SELECT m.name
FROM PEAK m, CLIMBED c
WHERE m.name = c.peak
GROUP BY m.NAME
HAVING COUNT(c.trip_id) = 1;
---------------------


--5
--SQL FOR QUESTION 5
---------------------
SELECT DISTINCT m.NAME
FROM CLIMBED c, PARTICIPATED p, PEAK m
WHERE c.TRIP_ID = p.trip_id AND
     p.NAME = "JOHN" AND
     c.PEAK = m.NAME AND
     m.ELEV > 14000;
---------------------


--6
--SQL FOR QUESTION 6
---------------------
SELECT m.map
FROM PEAK m
GROUP BY m.map
HAVING MAX(m.ELEV) - MIN(m.ELEV) > 2000;
---------------------


--7
```

```
--SQL FOR QUESTION 7
--------------------
SELECT MAP, AVG(ELEV) AVG_ELEV
FROM PEAK
GROUP BY MAP
ORDER BY AVG_ELEV DESC;
--------------------


--8
--SQL FOR QUESTION 8
--------------------
CREATE VIEW MARK AS
SELECT DISTINCT c2.peak
FROM PARTICIPATED p2, climbed c2
WHERE p2.name = "MARK" AND
    p2.trip_id = c2.trip_id;

SELECT DISTINCT c.peak
FROM PARTICIPATED p, climbed c, MARK m
WHERE p.name = "MARY" AND
    p.trip_id = c.trip_id AND
    c.peak = m.peak;

DROP VIEW MARK;
--------------------


--9
--SQL FOR QUESTION 9
--------------------
CREATE VIEW PEAKS_CLIMBED AS
SELECT DISTINCT m.NAME
FROM PEAK m, CLIMBED c
WHERE m.NAME = c.PEAK;

CREATE VIEW PEAKS_UNCLIMBED AS
SELECT NAME
FROM PEAK
EXCEPT -- SQLite's version of the minus operator
SELECT name
FROM PEAKS_CLIMBED;

SELECT REGION, count(m2.NAME) AS NUM_UNCLIMBED
FROM PEAK m2, PEAKS_UNCLIMBED pu
WHERE m2.NAME = pu.NAME
GROUP BY REGION;

DROP VIEW PEAKS_UNCLIMBED;
DROP VIEW PEAKS_CLIMBED;
--------------------


--10
--SQL FOR QUESTION 10
--------------------
SELECT c.TRIP_ID
FROM CLIMBED c, PARTICIPATED p, PEAK m
WHERE c.TRIP_ID = p.TRIP_ID AND
        m.name = c.PEAK
GROUP BY c.trip_id
HAVING sum(ELEV) > 500000;
--------------------


--11
--SQL FOR QUESTION 11
--------------------
CREATE VIEW PEAK_COUNT_BY_SEX AS
```

```
SELECT c1.SEX, COUNT(p.TRIP_ID) PEAK_COUNT
FROM CLIMBER c1, PARTICIPATED p, CLIMBED c
WHERE c1.NAME = p.NAME AND
        c.TRIP_ID = p.TRIP_ID
GROUP BY c1.SEX;

CREATE VIEW CLIMBER_COUNT_BY_SEX as
SELECT SEX, COUNT(NAME) CLIMBER_COUNT
FROM CLIMBER c1
GROUP BY SEX;

SELECT ccbs.SEX, CAST(PEAK_COUNT AS FLOAT)/CLIMBER_COUNT AVG_PEAKS
FROM CLIMBER_COUNT_BY_SEX ccbs, PEAK_COUNT_BY_SEX pcbs
WHERE ccbs.SEX = pcbs.SEX;

DROP VIEW CLIMBER_COUNT_BY_SEX;
DROP VIEW PEAK_COUNT_BY_SEX;
----------------------


--12
--SQL FOR QUESTION 12
----------------------
CREATE VIEW MARIA AS
SELECT DISTINCT c.peak
FROM PARTICIPATED p, climbed c
WHERE p.name = "MARIA" AND
    p.trip_id = c.trip_id;

CREATE VIEW OTHERS AS
SELECT DISTINCT p.NAME, c.PEAK
FROM PARTICIPATED p, CLIMBED c
WHERE p.TRIP_ID = c.TRIP_ID;

SELECT NAME
FROM MARIA m, OTHERS o
WHERE m.PEAK = o.PEAK AND
        o.NAME != "MARIA"
GROUP BY o.NAME
HAVING COUNT(o.PEAK) = (SELECT count(m2.PEAK) FROM MARIA m2);

DROP VIEW MARIA;
DROP VIEW OTHERS;
----------------------


--13
--SQL FOR QUESTION 13
----------------------
CREATE VIEW TOTAL_PEAKS_CLIMBED_BY_REGION AS
SELECT REGION, count(NAME) AS TOTAL_PEAKS_CLIMBED
FROM PEAK m2
WHERE m2.NAME IN (SELECT DISTINCT m.NAME
                                FROM PEAK m, CLIMBED c
                                WHERE m.NAME = c.PEAK)
GROUP BY REGION;

CREATE VIEW TOTAL_PEAKS_BY_REGION AS
SELECT REGION, count(NAME) AS TOTAL_PEAKS
FROM PEAK m
GROUP BY REGION;

SELECT REGION, MAX(FRACTION_CLIMBED) LARGEST_FRACTION_CLIMBED
FROM (SELECT tp.REGION, CAST(TOTAL_PEAKS_CLIMBED AS FLOAT)/TOTAL_PEAKS FRACTION_CLIMBED
        FROM TOTAL_PEAKS_BY_REGION tp, TOTAL_PEAKS_CLIMBED_BY_REGION pc
        WHERE tp.REGION = pc.REGION
        GROUP BY tp.REGION);

DROP VIEW TOTAL_PEAKS_BY_REGION;
```

```
DROP VIEW TOTAL_PEAKS_CLIMBED_BY_REGION;
--------------------


--14
--SQL FOR QUESTION 14
--------------------
CREATE VIEW GROUPS AS
SELECT p1.NAME NAME1, p2.NAME NAME2, p1.TRIP_ID TRIP_ID
FROM PARTICIPATED p1, PARTICIPATED p2
WHERE p1.TRIP_ID = p2.TRIP_ID AND
    p1.NAME < p2.NAME; -- get unique pairs who went on the same trip

SELECT NAME1, NAME2, MAX(PEAKS_CLIMBED) PEAKS_CLIMBED
FROM (SELECT NAME1, NAME2, COUNT(c.PEAK) PEAKS_CLIMBED
    FROM GROUPS g, CLIMBED c
    WHERE g.TRIP_ID = c.TRIP_ID
    GROUP BY NAME1, NAME2);

DROP VIEW GROUPS;
--------------------
```