

Input

Run SQL

Available Tables

```
);
INSERT INTO STATION VALUES (1, 'ABCA', 'UP', 27.18, 78.02);
INSERT INTO STATION VALUES (2, 'DEF', 'PB', 31.63, 74.87);
INSERT INTO STATION VALUES (3, 'ADEF', 'MH', 19.88, 75.32);
INSERT INTO STATION VALUES (4, 'GHI', 'MP', 22.72, 75.86);
INSERT INTO STATION VALUES (5, 'FJK', 'KA', 13.34, 74.75);
INSERT INTO STATION VALUES (6, 'XYZ', 'GJ', 23.03, 72.58);
INSERT INTO STATION VALUES (7, 'UHA', 'MP', 22.61, 77.76);
INSERT INTO STATION VALUES (8, 'ABABA', 'UP', 25.45, 81.85);
INSERT INTO STATION VALUES (9, 'ILALA', 'TN', 11.41, 76.70);
INSERT INTO STATION VALUES (10, 'YUIK', 'KL', 9.98, 76.28);

SELECT DISTINCT CITY
FROM STATION
WHERE LOWER(SUBSTR(CITY, 1, 1)) IN ('a', 'e', 'i', 'o', 'u')
      AND LOWER(SUBSTR(CITY, LENGTH(CITY), 1)) IN ('a', 'e', 'i', 'o', 'u');
```

Output

CITY
ABCA
UHA
ABABA
ILALA

STATION

ID	CITY	STATE	LAT_N	LONG_W
1	ABCA	UP	27.18	78.02
2	DEF	PB	31.63	74.87
3	ADEF	MH	19.88	75.32
4	GHI	MP	22.72	75.86
5	FJK	KA	13.34	74.75
6	XYZ	GJ	23.03	72.58
7	UHA	MP	22.61	77.76
8	ABABA	UP	25.45	81.85
9	ILALA	TN	11.41	76.7
10	YUIK	KL	9.98	76.28

Input

Run SQL

>

Available Tables

```
CREATE TABLE CITY (  
  ID INT,  
  NAME VARCHAR(17),  
  COUNTRYCODE VARCHAR(3),  
  DISTRICT VARCHAR(20),  
  POPULATION INT  
);  
INSERT INTO CITY VALUES (1, 'Delhi', 'IND', 'Delhi', 19000000);  
INSERT INTO CITY VALUES (2, 'Tokyo', 'JPN', 'Tokyo', 14000000);  
INSERT INTO CITY VALUES (3, 'Paris', 'FRA', 'France', 2200000);  
INSERT INTO CITY VALUES (4, 'New York', 'USA', 'New York', 8500000);  
INSERT INTO CITY VALUES (5, 'Cairo', 'EGY', 'Cairo', 9900000);  
select max(population)-min(population) FROM CITY;
```

CITY

ID	NAME	COUNTRYCODE	DISTRICT	POPULATION
1	Delhi	IND	Delhi	19000000
2	Tokyo	JPN	Tokyo	14000000
3	Paris	FRA	France	2200000
4	New York	USA	New York	8500000
5	Cairo	EGY	Cairo	9900000

Output

max(population)-min(population)
16800000

Input

Run SQL

> Available Tables

LONG\_W FLOAT

```
);
INSERT INTO STATION VALUES (1, 'ABCA', 'UP', 27.18, 78.02);
INSERT INTO STATION VALUES (2, 'DEF', 'PB', 31.63, 74.87);
INSERT INTO STATION VALUES (3, 'ADEF', 'MH', 19.88, 75.32);
INSERT INTO STATION VALUES (4, 'GHI', 'MP', 22.72, 75.86);
INSERT INTO STATION VALUES (5, 'FJK', 'KA', 13.34, 74.75);
INSERT INTO STATION VALUES (6, 'XYZ', 'GJ', 23.03, 72.58);
INSERT INTO STATION VALUES (7, 'UHA', 'MP', 22.61, 77.76);
INSERT INTO STATION VALUES (8, 'ABABA', 'UP', 25.45, 81.85);
INSERT INTO STATION VALUES (9, 'ILALA', 'TN', 11.41, 76.70);
INSERT INTO STATION VALUES (10, 'YUIK', 'KL', 9.98, 76.28);

SELECT CAST(SQRT((MAX(LAT_N) - MIN(LAT_N)) * (MAX(LAT_N) - MIN(LAT_N)) + (MAX(LONG_W) - MIN(LONG_W)) * (MAX(LONG_W) - MIN(LONG_W))) AS DECIMAL(10, 4)) AS distance FROM STATION;
```

Output

distance

23.55112311546946

STATION

ID	CITY	STATE	LAT_N	LONG_W
1	ABCA	UP	27.18	78.02
2	DEF	PB	31.63	74.87
3	ADEF	MH	19.88	75.32
4	GHI	MP	22.72	75.86
5	FJK	KA	13.34	74.75
6	XYZ	GJ	23.03	72.58
7	UHA	MP	22.61	77.76
8	ABABA	UP	25.45	81.85
9	ILALA	TN	11.41	76.7
10	YUIK	KL	9.98	76.28

Input

```
INSERT INTO STATION VALUES (1, 'ABCA', 'UP', 27.18, 78.02);
INSERT INTO STATION VALUES (2, 'DEF', 'PB', 31.63, 74.87);
INSERT INTO STATION VALUES (3, 'ADEF', 'MH', 19.88, 75.32);
INSERT INTO STATION VALUES (4, 'GHI', 'MP', 22.72, 75.86);
INSERT INTO STATION VALUES (5, 'FJK', 'KA', 13.34, 74.75);
INSERT INTO STATION VALUES (6, 'XYZ', 'GJ', 23.03, 72.58);
INSERT INTO STATION VALUES (7, 'UHA', 'MP', 22.61, 77.76);
INSERT INTO STATION VALUES (8, 'ABABA', 'UP', 25.45, 81.85);
INSERT INTO STATION VALUES (9, 'ILALA', 'TN', 11.41, 76.70);
INSERT INTO STATION VALUES (10, 'YUIK', 'KL', 9.98, 76.28);

SELECT ROUND(lat_n, 4)
FROM (SELECT LAT_N, RANK() OVER(ORDER BY lat_n ASC) AS ranked FROM station) AS
station_temp
WHERE ranked = (SELECT ROUND(COUNT(*) / 2) FROM station)
```

Run SQL

Available Tables

STATION

ID	CITY	STATE	LAT_N	LONG_W
1	ABCA	UP	27.18	78.02
2	DEF	PB	31.63	74.87
3	ADEF	MH	19.88	75.32
4	GHI	MP	22.72	75.86
5	FJK	KA	13.34	74.75
6	XYZ	GJ	23.03	72.58
7	UHA	MP	22.61	77.76
8	ABABA	UP	25.45	81.85
9	ILALA	TN	11.41	76.7
10	YUIK	KL	9.98	76.28

Output

ROUND(lat\_n, 4)

22.61

Input

Run SQL

Available Tables

POPULATION NUMBER

```
);
INSERT INTO CITY VALUES (1, 'Mumbai', 'IND', 'Maharashtra', 20411000);
INSERT INTO CITY VALUES (2, 'Delhi', 'IND', 'Delhi', 16787941);
INSERT INTO CITY VALUES (3, 'New York', 'USA', 'New York', 8419600);
INSERT INTO CITY VALUES (4, 'Los Angeles', 'USA', 'California', 3980400);
INSERT INTO CITY VALUES (5, 'Paris', 'FRA', 'France', 2148271);
INSERT INTO CITY VALUES (6, 'Lagos', 'NGA', 'Lagos', 14000000);
INSERT INTO CITY VALUES (7, 'Abuja', 'NGA', 'Federal Capital', 1235880);

SELECT ci.NAME
FROM CITY ci
JOIN COUNTRY co ON ci.COUNTRYCODE = co.CODE
WHERE co.CONTINENT = 'Africa';
```

Output

NAME
Abuja
Lagos

		IND	Maharashtra	20411000
2	Delhi	IND	Delhi	16787941
3	New York	USA	New York	8419600
4	Los Angeles	USA	California	3980400
5	Paris	FRA	Ile-de-France	2148271
6	Lagos	NGA	Lagos	14000000
7	Abuja	NGA	Federal Capital	1235880

COUNTRY

CODE	NAME	CONTINENT	REGION	SURFACEAREA
IND	India	Asia	Southern Asia	3287263
USA	United States	North America	North America	9833520
NGA	Nigeria	Africa	Western Africa	923768
FRA	France	Europe	Western Europe	551695

Input

POPULATION NUMBER

);

INSERT INTO CITY VALUES (1, 'Mumbai', 'IND', 'Maharashtra', 20411000);

INSERT INTO CITY VALUES (2, 'Delhi', 'IND', 'Delhi', 16787941);

INSERT INTO CITY VALUES (3, 'New York', 'USA', 'New York', 8419600);

INSERT INTO CITY VALUES (4, 'Los Angeles', 'USA', 'California', 3980400);

INSERT INTO CITY VALUES (5, 'Paris', 'FRA', 'France', 2148271);

INSERT INTO CITY VALUES (6, 'Lagos', 'NGA', 'Lagos', 14000000);

INSERT INTO CITY VALUES (7, 'Abuja', 'NGA', 'Federal Capital', 1235880);

SELECT ci.NAME

FROM CITY ci

JOIN COUNTRY co ON ci.COUNTRYCODE = co.CODE

WHERE co.CONTINENT = 'Africa';

Run SQL

Available Tables

Output

NAME
Abuja
Lagos

COUNTRY

CODE	NAME	CONTINENT	REGION	SURFACEAREA
IND	India	Asia	Southern Asia	3287263
USA	United States	North America	North America	9833520
NGA	Nigeria	Africa	Western Africa	923768
FRA	France	Europe	Western Europe	551695

Input

Run SQL

```
        WHEN g.Grade < 8 THEN 'NULL'
        ELSE s.Name
    END AS Name,
    g.Grade,
    s.Marks
FROM Students s
JOIN Grades g
    ON s.Marks BETWEEN g.Min_Mark AND g.Max_Mark
ORDER BY
    g.Grade DESC,
    CASE
        WHEN g.Grade < 8 THEN s.Marks
        ELSE s.Name
    END;
```

Available Tables

Grade	Min_Mark	Max_Mark
1	0	9
2	10	19
3	20	29
4	30	39
5	40	49
6	50	59
7	60	69
8	70	79
9	80	89
10	90	100

ID	Name	Marks
1	Julia	88
2	Samantha	68
3	Maria	99
4	Scarlet	78
5	Ashley	63
6	Jane	81

Output

Name	Grade	Marks
Maria	10	99
Jane	9	81
Julia	9	88
Scarlet	8	78
NULL	7	63
NULL	7	68

Input

```
(3, 3, 102, 100),
(4, 2, 103, 70),
(5, 4, 103, 70),
(6, 1, 104, 30);

SELECT
  h.hacker_id,
  h.name
FROM Submissions s
JOIN Challenges c ON s.challenge_id = c.challenge_id
JOIN Difficulty d ON c.difficulty_level = d.difficulty_level
JOIN Hackers h ON s.hacker_id = h.hacker_id
WHERE s.score = d.score
GROUP BY h.hacker_id, h.name
HAVING COUNT(*) > 1
ORDER BY COUNT(*) DESC, h.hacker_id;
```

Run SQL

Available Tables

Grade	Min_Mark	Max_Mark
1	0	9
2	10	19
3	20	29
4	30	39
5	40	49
6	50	59
7	60	69
8	70	79
9	80	89
10	90	100

Hackers

hacker_id	name
101	Alice
102	Bob
103	Charlie
104	David

Students

Output

hacker_id	name
101	Alice
103	Charlie



Input

Run SQL

```
);

INSERT INTO Wands_Property VALUES
(1, 10, 0),
(2, 20, 0),
(3, 30, 1);

SELECT w.code, w.power, wp.age
FROM Wands w
JOIN Wands_Property wp ON w.code = wp.code AND w.age = wp.age
WHERE wp.is_evil = 0
AND w.power = (
    SELECT MAX(power)
    FROM Wands w2
    WHERE w2.code = w.code AND w2.age = w.age
)
ORDER BY w.power DESC, wp.age DESC;
```

Output

code	power	age
1	100	10
2	95	20

Available Tables

Wands

code	power	age
1	100	10
1	85	10
2	90	20
2	95	20
3	80	30

Wands\_Property

code	age	is_evil
1	10	0
2	20	0
3	30	1

Input

```
);
INSERT INTO Submissions VALUES
(1, 101, 1, 80),
(2, 101, 1, 90),
(3, 101, 2, 60),
(4, 102, 1, 30),
(5, 102, 2, 40),
(6, 103, 2, 0),
(7, 104, 3, 95),
(8, 104, 3, 95);

SELECT h.hacker_id, h.name , sum(s.marks) AS total
  FROM hackers h left join (SELECT hacker_id,challenge_id,max(score) as marks from
submissions
  group by hacker_id,challenge_id) s ON h.hacker_id=s.hacker_id
 GROUP BY h.hacker_id,h.name having total > 0
 ORDER BY total desc, h.hacker_id asc
```

Run SQL

Available Tables

Hackers

hacker_id	name
101	Alice
102	Bob
103	Charlie
104	David

Submissions

submission_id	hacker_id	challenge_id	score
1	101	1	80
2	101	1	90
3	101	2	60
4	102	1	30
5	102	2	40
6	103	2	0
7	104	3	95
8	104	3	95

Wands

wand_id	hacker_id	challenge_id	score
---------	-----------	--------------	-------

Output

hacker_id	name	total
101	Alice	150
104	David	95
102	Bob	70