|  |  |
| --- | --- |
|  | --Q51: |
|  |  |
|  | CREATE TABLE world |
|  | ( |
|  | name VARCHAR(25), |
|  | continent VARCHAR(10), |
|  | area INT, |
|  | population INT, |
|  | gdp BIGINT, |
|  | CONSTRAINT pk\_world PRIMARY KEY (name) |
|  | ); |
|  |  |
|  | INSERT INTO world VALUES('Afghanistan', 'Asia', 652230, 25500100, 20343000000); |
|  | INSERT INTO world VALUES('Albania', 'Europe', 28748, 2831741, 12960000000); |
|  | INSERT INTO world VALUES('Algeria', 'Africa', 2381741, 37100000, 188681000000); |
|  | INSERT INTO world VALUES('Andorra', 'Europe', 468, 78115, 3712000000); |
|  | INSERT INTO world VALUES('Angola', 'Africa', 1246700, 20609294, 100990000000); |
|  |  |
|  | SELECT |
|  | name, |
|  | population, |
|  | area |
|  | FROM |
|  | world |
|  | WHERE |
|  | area >= 3000000 |
|  | OR population >= 25000000 |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE world; |
|  |  |
|  |  |
|  | --Q52: |
|  |  |
|  | CREATE TABLE customer |
|  | ( |
|  | id INT, |
|  | name VARCHAR(25), |
|  | referee\_id BIGINT, |
|  | CONSTRAINT pk\_customer PRIMARY KEY (id) |
|  | ); |
|  |  |
|  | INSERT INTO customer VALUES(1, 'Will', null); |
|  | INSERT INTO customer VALUES(2, 'Jane', null); |
|  | INSERT INTO customer VALUES(3, 'Alex', 2); |
|  | INSERT INTO customer VALUES(4, 'Bill', null); |
|  | INSERT INTO customer VALUES(5, 'Zack', 1); |
|  | INSERT INTO customer VALUES(6, 'Mark', 2); |
|  |  |
|  | SELECT |
|  | name |
|  | FROM |
|  | customer |
|  | WHERE |
|  | referee\_id <> 2 |
|  | OR referee\_id IS NULL |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE customer; |
|  |  |
|  |  |
|  | --Q53: |
|  |  |
|  | CREATE TABLE customers |
|  | ( |
|  | id INT, |
|  | name VARCHAR(25), |
|  | CONSTRAINT pk\_customers PRIMARY KEY (id) |
|  | ); |
|  |  |
|  | CREATE TABLE orders |
|  | ( |
|  | id INT, |
|  | customer\_id INT, |
|  | CONSTRAINT pk\_orders PRIMARY KEY (id), |
|  | CONSTRAINT fk\_customer\_order FOREIGN KEY(customer\_id) |
|  | REFERENCES customers(id) |
|  | ); |
|  |  |
|  | INSERT INTO customers VALUES(1, 'Joe'); |
|  | INSERT INTO customers VALUES(2, 'Henry'); |
|  | INSERT INTO customers VALUES(3, 'Sam'); |
|  | INSERT INTO customers VALUES(4, 'Max'); |
|  |  |
|  | INSERT INTO orders VALUES(1, 3); |
|  | INSERT INTO orders VALUES(2, 1); |
|  |  |
|  | SELECT |
|  | name |
|  | FROM |
|  | customers c |
|  | WHERE |
|  | NOT EXISTS( |
|  | SELECT |
|  | \* |
|  | FROM |
|  | orders o |
|  | WHERE c.id = o.customer\_id |
|  | ) |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE orders; |
|  | DROP TABLE customers; |
|  |  |
|  |  |
|  | --Q54: |
|  |  |
|  | CREATE TABLE employee |
|  | ( |
|  | employee\_id INT, |
|  | team\_id INT, |
|  | CONSTRAINT pk\_employee PRIMARY KEY (employee\_id) |
|  | ); |
|  |  |
|  | INSERT INTO employee VALUES(1, 8); |
|  | INSERT INTO employee VALUES(2, 8); |
|  | INSERT INTO employee VALUES(3, 8); |
|  | INSERT INTO employee VALUES(4, 7); |
|  | INSERT INTO employee VALUES(5, 9); |
|  | INSERT INTO employee VALUES(6, 9); |
|  |  |
|  | SELECT |
|  | employee\_id, |
|  | count(\*) OVER(PARTITION BY team\_id) AS team\_size |
|  | FROM |
|  | employee |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE employee; |
|  |  |
|  |  |
|  | --Q55: |
|  |  |
|  | CREATE TABLE person |
|  | ( |
|  | id INT, |
|  | name VARCHAR(25), |
|  | phone\_number VARCHAR(11), |
|  | CONSTRAINT pk\_person PRIMARY KEY (id) |
|  | ); |
|  |  |
|  | CREATE TABLE country |
|  | ( |
|  | name VARCHAR(25), |
|  | country\_code VARCHAR(3), |
|  | CONSTRAINT pk\_country PRIMARY KEY (country\_code) |
|  | ); |
|  |  |
|  | CREATE TABLE calls |
|  | ( |
|  | caller\_id INT, |
|  | callee\_id INT, |
|  | duration INT |
|  | ); |
|  |  |
|  | INSERT INTO person VALUES(3, 'Jonathan', '051-1234567'); |
|  | INSERT INTO person VALUES(12, 'Elvis', '051-7654321'); |
|  | INSERT INTO person VALUES(1, 'Moncef', '212-1234567'); |
|  | INSERT INTO person VALUES(2, 'Maroua', '212-6523651'); |
|  | INSERT INTO person VALUES(7, 'Meir', '972-1234567'); |
|  | INSERT INTO person VALUES(9, 'Rachel', '972-0011100'); |
|  |  |
|  | INSERT INTO country VALUES('Peru', '51'); |
|  | INSERT INTO country VALUES('Israel', '972'); |
|  | INSERT INTO country VALUES('Morocco', '212'); |
|  | INSERT INTO country VALUES('Germany', '49'); |
|  | INSERT INTO country VALUES('Ethiopia', '251'); |
|  |  |
|  | INSERT INTO calls VALUES(1, 9, 33); |
|  | INSERT INTO calls VALUES(2, 9, 4); |
|  | INSERT INTO calls VALUES(1, 2, 59); |
|  | INSERT INTO calls VALUES(3, 12, 102); |
|  | INSERT INTO calls VALUES(3, 12, 330); |
|  | INSERT INTO calls VALUES(12, 3, 5); |
|  | INSERT INTO calls VALUES(7, 9, 13); |
|  | INSERT INTO calls VALUES(7, 1, 3); |
|  | INSERT INTO calls VALUES(9, 7, 1); |
|  | INSERT INTO calls VALUES(1, 7, 7); |
|  |  |
|  | WITH receiver\_caller\_calls AS( |
|  | SELECT |
|  | caller\_id AS caller\_receiver\_id, |
|  | duration |
|  | FROM |
|  | calls |
|  | UNION ALL |
|  | SELECT |
|  | callee\_id AS caller\_receiver\_id, |
|  | duration |
|  | FROM |
|  | calls |
|  | ), |
|  | call\_duration\_avg AS( |
|  | SELECT |
|  | DISTINCT cn.name, |
|  | avg(c.duration) OVER() as global\_average, |
|  | avg(c.duration) OVER(PARTITION BY cn.name) as country\_average |
|  | FROM |
|  | person p |
|  | JOIN country cn |
|  | ON CAST(SUBSTRING\_INDEX(p.phone\_number, '-', 1) AS UNSIGNED) = CAST(cn.country\_code AS UNSIGNED) |
|  | JOIN receiver\_caller\_calls c |
|  | ON c.caller\_receiver\_id = p.id |
|  | ) |
|  | SELECT |
|  | name |
|  | FROM |
|  | call\_duration\_avg |
|  | WHERE |
|  | country\_average > global\_average |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE person; |
|  | DROP TABLE country; |
|  | DROP TABLE calls; |
|  |  |
|  |  |
|  | --Q56: |
|  |  |
|  | CREATE TABLE activity |
|  | ( |
|  | player\_id INT, |
|  | device\_id INT, |
|  | event\_date DATE, |
|  | games\_played INT |
|  | ); |
|  |  |
|  | INSERT INTO activity VALUES(1, 2, '2016-03-01', 5); |
|  | INSERT INTO activity VALUES(1, 2, '2016-05-02', 6); |
|  | INSERT INTO activity VALUES(2, 3, '2017-06-25', 1); |
|  | INSERT INTO activity VALUES(3, 1, '2016-03-02', 0); |
|  | INSERT INTO activity VALUES(3, 4, '2018-07-03', 5); |
|  |  |
|  | WITH activity\_with\_device\_serial AS( |
|  | SELECT |
|  | player\_id, |
|  | device\_id, |
|  | DENSE\_RANK() OVER(PARTITION BY player\_id ORDER BY event\_date) as device\_serial |
|  | FROM |
|  | activity |
|  | ) |
|  | SELECT |
|  | player\_id, |
|  | device\_id |
|  | FROM |
|  | activity\_with\_device\_serial |
|  | WHERE |
|  | device\_serial = 1 |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE activity; |
|  |  |
|  |  |
|  | --Q57: |
|  |  |
|  | CREATE TABLE orders |
|  | ( |
|  | order\_number INT, |
|  | customer\_number INT, |
|  | CONSTRAINT pk\_orders PRIMARY KEY (order\_number) |
|  | ); |
|  |  |
|  | INSERT INTO orders VALUES(1, 1); |
|  | INSERT INTO orders VALUES(2, 2); |
|  | INSERT INTO orders VALUES(3, 3); |
|  | INSERT INTO orders VALUES(4, 3); |
|  |  |
|  | WITH customer\_with\_no\_of\_order AS( |
|  | SELECT |
|  | customer\_number, |
|  | count(\*) AS no\_of\_order |
|  | FROM |
|  | orders |
|  | GROUP BY customer\_number |
|  | ) |
|  | SELECT |
|  | customer\_number |
|  | FROM |
|  | customer\_with\_no\_of\_order |
|  | ORDER BY no\_of\_order DESC |
|  | LIMIT 1; |
|  |  |
|  | --Follow Up: |
|  |  |
|  | INSERT INTO orders VALUES(5, 2); |
|  |  |
|  | WITH customer\_with\_no\_of\_order AS( |
|  | SELECT |
|  | customer\_number, |
|  | count(\*) AS no\_of\_order |
|  | FROM |
|  | orders |
|  | GROUP BY customer\_number |
|  | ) |
|  | SELECT |
|  | customer\_number |
|  | FROM |
|  | customer\_with\_no\_of\_order |
|  | WHERE |
|  | no\_of\_order IN ( |
|  | SELECT |
|  | max(no\_of\_order) AS max\_no\_of\_order |
|  | FROM |
|  | customer\_with\_no\_of\_order |
|  | ) |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE orders; |
|  |  |
|  |  |
|  | --Q58: |
|  |  |
|  | CREATE TABLE cinema |
|  | ( |
|  | seat\_id INT AUTO\_INCREMENT, |
|  | free BOOLEAN, |
|  | CONSTRAINT pk\_cinema PRIMARY KEY (seat\_id) |
|  | ); |
|  |  |
|  | INSERT INTO cinema(free) VALUES(TRUE); |
|  | INSERT INTO cinema(free) VALUES(FALSE); |
|  | INSERT INTO cinema(free) VALUES(TRUE); |
|  | INSERT INTO cinema(free) VALUES(TRUE); |
|  | INSERT INTO cinema(free) VALUES(TRUE); |
|  |  |
|  | WITH seat\_id\_with\_diff\_with\_next\_val AS ( |
|  | SELECT |
|  | seat\_id, |
|  | seat\_id - LEAD(seat\_id, 1) OVER(ORDER BY seat\_id) AS diff\_next\_val, |
|  | seat\_id - LAG(seat\_id, 1) OVER(ORDER BY seat\_id) AS diff\_prev\_val |
|  | FROM |
|  | cinema |
|  | WHERE |
|  | free <> 0 |
|  | ) |
|  | SELECT |
|  | seat\_id |
|  | FROM |
|  | seat\_id\_with\_diff\_with\_next\_val |
|  | WHERE |
|  | diff\_next\_val = -1 |
|  | OR diff\_prev\_val = 1 |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE cinema; |
|  |  |
|  |  |
|  | --Q59: |
|  |  |
|  | CREATE TABLE sales\_person |
|  | ( |
|  | sales\_id INT, |
|  | name VARCHAR(25), |
|  | salary INT, |
|  | commission\_rate INT, |
|  | hire\_rate DATE, |
|  | CONSTRAINT pk\_sales\_person PRIMARY KEY (sales\_id) |
|  | ); |
|  |  |
|  | CREATE TABLE company |
|  | ( |
|  | com\_id INT, |
|  | name VARCHAR(25), |
|  | city VARCHAR(25), |
|  | CONSTRAINT pk\_company PRIMARY KEY (com\_id) |
|  | ); |
|  |  |
|  | CREATE TABLE orders |
|  | ( |
|  | order\_id INT, |
|  | order\_date DATE, |
|  | com\_id INT, |
|  | sales\_id INT, |
|  | amount INT, |
|  | CONSTRAINT pk\_orders PRIMARY KEY (order\_id), |
|  | CONSTRAINT fk\_company FOREIGN KEY (com\_id) |
|  | REFERENCES company(com\_id), |
|  | CONSTRAINT fk\_sales\_person FOREIGN KEY (sales\_id) |
|  | REFERENCES sales\_person(sales\_id) |
|  | ); |
|  |  |
|  |  |
|  | INSERT INTO sales\_person VALUES(1, 'John', 100000, 6, '2006-04-01'); |
|  | INSERT INTO sales\_person VALUES(2, 'Amy', 12000, 5, '2010-05-01'); |
|  | INSERT INTO sales\_person VALUES(3, 'Mark', 65000, 12, '2008-12-25'); |
|  | INSERT INTO sales\_person VALUES(4, 'Pam', 25000, 25, '2005-01-01'); |
|  | INSERT INTO sales\_person VALUES(5, 'Alex', 5000, 10, '2007-02-03'); |
|  |  |
|  | INSERT INTO company VALUES(1, 'RED', 'Boston'); |
|  | INSERT INTO company VALUES(2, 'ORANGE', 'New York'); |
|  | INSERT INTO company VALUES(3, 'YELLOW', 'Boston'); |
|  | INSERT INTO company VALUES(4, 'GREEN', 'Austin'); |
|  |  |
|  |  |
|  | INSERT INTO orders VALUES(1, '2014-01-01', 3, 4, 10000); |
|  | INSERT INTO orders VALUES(2, '2014-02-01', 4, 5, 5000); |
|  | INSERT INTO orders VALUES(3, '2014-03-01', 1, 1, 50000); |
|  | INSERT INTO orders VALUES(4, '2014-04-01', 1, 4, 25000); |
|  |  |
|  | SELECT |
|  | name |
|  | FROM |
|  | sales\_person sp |
|  | WHERE |
|  | NOT EXISTS( |
|  | SELECT |
|  | \* |
|  | FROM |
|  | orders o |
|  | JOIN company c ON o.com\_id = c.com\_id |
|  | WHERE |
|  | sp.sales\_id = o.sales\_id |
|  | AND c.name = 'RED' |
|  | ) |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE orders; |
|  | DROP TABLE company; |
|  | DROP TABLE sales\_person; |
|  |  |
|  |  |
|  | --Q60: |
|  |  |
|  | CREATE TABLE triangle |
|  | ( |
|  | x INT, |
|  | y INT, |
|  | z INT, |
|  | CONSTRAINT pk\_triangle PRIMARY KEY (x,y,z) |
|  | ); |
|  |  |
|  | INSERT INTO triangle VALUES(13, 15, 30); |
|  | INSERT INTO triangle VALUES(10, 20, 15); |
|  |  |
|  | SELECT |
|  | \*, |
|  | CASE |
|  | WHEN x + y > z AND y + z > x AND z + x > y |
|  | THEN 'Yes' |
|  | ELSE 'No' |
|  | END as triangle |
|  | FROM |
|  | triangle |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE triangle; |
|  |  |
|  |  |
|  | --Q61: |
|  |  |
|  | CREATE TABLE point |
|  | ( |
|  | x INT, |
|  | CONSTRAINT pk\_point PRIMARY KEY (x) |
|  | ); |
|  |  |
|  | INSERT INTO point VALUES(-1); |
|  | INSERT INTO point VALUES(0); |
|  | INSERT INTO point VALUES(2); |
|  |  |
|  | SELECT |
|  | MIN(ABS(p1.x - p2.x)) AS shortest |
|  | FROM |
|  | point p1 |
|  | JOIN point p2 ON p1.x <> p2.x |
|  | ; |
|  |  |
|  | --Follow Up: |
|  |  |
|  | WITH ordered\_point AS( |
|  | SELECT |
|  | x, |
|  | ROW\_NUMBER() OVER(ORDER BY x) AS ordered\_no |
|  | FROM |
|  | point |
|  | ) |
|  | SELECT |
|  | MIN(ABS(p1.x - p2.x)) AS shortest |
|  | FROM |
|  | ordered\_point p1 |
|  | JOIN ordered\_point p2 ON p2.ordered\_no > p1.ordered\_no |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE point; |
|  |  |
|  |  |
|  | --Q62: |
|  |  |
|  | CREATE TABLE actor\_director |
|  | ( |
|  | actor\_id INT, |
|  | director\_id INT, |
|  | timestamp INT, |
|  | CONSTRAINT pk\_actor\_director PRIMARY KEY (timestamp) |
|  | ); |
|  |  |
|  | INSERT INTO actor\_director VALUES(1, 1, 0); |
|  | INSERT INTO actor\_director VALUES(1, 1, 1); |
|  | INSERT INTO actor\_director VALUES(1, 1, 2); |
|  | INSERT INTO actor\_director VALUES(1, 2, 3); |
|  | INSERT INTO actor\_director VALUES(1, 2, 4); |
|  | INSERT INTO actor\_director VALUES(2, 1, 5); |
|  | INSERT INTO actor\_director VALUES(2, 1, 6); |
|  |  |
|  | SELECT |
|  | actor\_id, |
|  | director\_id |
|  | FROM |
|  | actor\_director |
|  | GROUP BY |
|  | actor\_id, |
|  | director\_id |
|  | HAVING |
|  | count(\*) >= 3 |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE actor\_director; |
|  |  |
|  |  |
|  | --Q63: |
|  |  |
|  | CREATE TABLE product |
|  | ( |
|  | product\_id INT, |
|  | product\_name VARCHAR(25), |
|  | CONSTRAINT pk\_product PRIMARY KEY (product\_id) |
|  | ); |
|  |  |
|  | CREATE TABLE sales |
|  | ( |
|  | sale\_id INT, |
|  | product\_id INT, |
|  | year INT, |
|  | quantity INT, |
|  | price INT, |
|  | CONSTRAINT pk\_sales PRIMARY KEY (sale\_id, year), |
|  | CONSTRAINT fk\_product FOREIGN KEY (product\_id) |
|  | REFERENCES product(product\_id) |
|  | ); |
|  |  |
|  | INSERT INTO product VALUES(100, 'Nokia'); |
|  | INSERT INTO product VALUES(200, 'Apple'); |
|  | INSERT INTO product VALUES(300, 'Samsung'); |
|  |  |
|  | INSERT INTO sales VALUES(1, 100, 2008, 10, 5000); |
|  | INSERT INTO sales VALUES(2, 100, 2009, 12, 5000); |
|  | INSERT INTO sales VALUES(7, 200, 2011, 15, 9000); |
|  |  |
|  | SELECT |
|  | product\_name, |
|  | year, |
|  | price |
|  | FROM |
|  | product p |
|  | JOIN sales s ON p.product\_id = s.product\_id |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE sales; |
|  | DROP TABLE product; |
|  |  |
|  |  |
|  | --Q64: |
|  |  |
|  | CREATE TABLE employee |
|  | ( |
|  | employee\_id INT, |
|  | name VARCHAR(25), |
|  | experience\_years INT, |
|  | CONSTRAINT pk\_employee PRIMARY KEY (employee\_id) |
|  | ); |
|  |  |
|  | CREATE TABLE project |
|  | ( |
|  | project\_id INT, |
|  | employee\_id INT, |
|  | CONSTRAINT pk\_project PRIMARY KEY (project\_id, employee\_id), |
|  | CONSTRAINT fk\_employee FOREIGN KEY (employee\_id) |
|  | REFERENCES employee(employee\_id) |
|  | ); |
|  |  |
|  | INSERT INTO employee VALUES(1, 'Khaled', 3); |
|  | INSERT INTO employee VALUES(2, 'Ali', 2); |
|  | INSERT INTO employee VALUES(3, 'John', 1); |
|  | INSERT INTO employee VALUES(4, 'Doe', 2); |
|  |  |
|  | INSERT INTO project VALUES(1, 1); |
|  | INSERT INTO project VALUES(1, 2); |
|  | INSERT INTO project VALUES(1, 3); |
|  | INSERT INTO project VALUES(2, 1); |
|  | INSERT INTO project VALUES(2, 4); |
|  |  |
|  | SELECT |
|  | p.project\_id, |
|  | ROUND(avg(e.experience\_years),2) AS average\_years |
|  | FROM |
|  | project p |
|  | JOIN employee e ON p.employee\_id = e. employee\_id |
|  | GROUP BY p.project\_id |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE project; |
|  | DROP TABLE employee; |
|  |  |
|  |  |
|  | --Q65: |
|  |  |
|  | CREATE TABLE product |
|  | ( |
|  | product\_id INT, |
|  | product\_name VARCHAR(25), |
|  | unit\_price INT, |
|  | CONSTRAINT pk\_product PRIMARY KEY (product\_id) |
|  | ); |
|  |  |
|  | CREATE TABLE sales |
|  | ( |
|  | seller\_id INT, |
|  | product\_id INT, |
|  | buyer\_id INT, |
|  | sale\_date DATE, |
|  | quantity INT, |
|  | price INT, |
|  | CONSTRAINT fk\_product FOREIGN KEY (product\_id) |
|  | REFERENCES product(product\_id) |
|  | ); |
|  |  |
|  | INSERT INTO product VALUES(1, 'S8', 1000); |
|  | INSERT INTO product VALUES(2, 'G4', 800); |
|  | INSERT INTO product VALUES(3, 'iPhone', 1400); |
|  |  |
|  | INSERT INTO sales VALUES(1, 1, 1, '2019-01-21', 2, 2000); |
|  | INSERT INTO sales VALUES(1, 2, 2, '2019-02-17', 1, 800); |
|  | INSERT INTO sales VALUES(2, 2, 3, '2019-06-02', 1, 800); |
|  | INSERT INTO sales VALUES(3, 3, 4, '2019-05-13', 2, 2800); |
|  |  |
|  | WITH sales\_rank AS( |
|  | SELECT |
|  | seller\_id, |
|  | sum(price) AS total\_sales, |
|  | DENSE\_RANK() OVER(ORDER BY sum(price) DESC) rank\_by\_sales |
|  | FROM |
|  | sales |
|  | GROUP BY |
|  | seller\_id |
|  | ) |
|  | SELECT |
|  | seller\_id |
|  | FROM |
|  | sales\_rank |
|  | WHERE |
|  | rank\_by\_sales = 1 |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE sales; |
|  | DROP TABLE product; |
|  |  |
|  |  |
|  | --Q66: |
|  |  |
|  | CREATE TABLE product |
|  | ( |
|  | product\_id INT, |
|  | product\_name VARCHAR(25), |
|  | unit\_price INT, |
|  | CONSTRAINT pk\_product PRIMARY KEY (product\_id) |
|  | ); |
|  |  |
|  | CREATE TABLE sales |
|  | ( |
|  | seller\_id INT, |
|  | product\_id INT, |
|  | buyer\_id INT, |
|  | sale\_date DATE, |
|  | quantity INT, |
|  | price INT, |
|  | CONSTRAINT fk\_product FOREIGN KEY (product\_id) |
|  | REFERENCES product(product\_id) |
|  | ); |
|  |  |
|  | INSERT INTO product VALUES(1, 'S8', 1000); |
|  | INSERT INTO product VALUES(2, 'G4', 800); |
|  | INSERT INTO product VALUES(3, 'iPhone', 1400); |
|  |  |
|  | INSERT INTO sales VALUES(1, 1, 1, '2019-01-21', 2, 2000); |
|  | INSERT INTO sales VALUES(1, 2, 2, '2019-02-17', 1, 800); |
|  | INSERT INTO sales VALUES(2, 1, 3, '2019-06-02', 1, 800); |
|  | INSERT INTO sales VALUES(3, 3, 3, '2019-05-13', 2, 2800); |
|  |  |
|  | WITH s8\_iphone\_sales AS( |
|  | SELECT |
|  | s.buyer\_id, |
|  | p.product\_name |
|  | FROM |
|  | sales s |
|  | JOIN product p ON s.product\_id = p.product\_id |
|  | WHERE |
|  | p.product\_name = 'S8' |
|  | OR p.product\_name = 'iPhone' |
|  | ) |
|  |  |
|  | SELECT |
|  | s1.buyer\_id |
|  | FROM |
|  | s8\_iphone\_sales s1 |
|  | LEFT JOIN s8\_iphone\_sales s2 |
|  | ON (s1.product\_name = 'S8' |
|  | OR s1.product\_name = 'iPhone') |
|  | AND s2.product\_name = 'iPhone' |
|  | AND s1.buyer\_id = s2.buyer\_id |
|  | WHERE |
|  | s2.buyer\_id is null |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE sales; |
|  | DROP TABLE product; |
|  |  |
|  |  |
|  | --Q67: |
|  |  |
|  | CREATE TABLE customer |
|  | ( |
|  | customer\_id INT, |
|  | name VARCHAR(25), |
|  | visited\_on DATE, |
|  | amount INT, |
|  | CONSTRAINT pk\_customer PRIMARY KEY (customer\_id, visited\_on) |
|  | ); |
|  |  |
|  | INSERT INTO customer VALUES(1, 'Jhon', '2019-01-01', 100); |
|  | INSERT INTO customer VALUES(2, 'Daniel', '2019-01-02', 110); |
|  | INSERT INTO customer VALUES(3, 'Jade', '2019-01-03', 120); |
|  | INSERT INTO customer VALUES(4, 'Khaled', '2019-01-04', 130); |
|  | INSERT INTO customer VALUES(5, 'Winston', '2019-01-05', 110); |
|  | INSERT INTO customer VALUES(6, 'Elvis', '2019-01-06', 140); |
|  | INSERT INTO customer VALUES(7, 'Anna', '2019-01-07', 150); |
|  | INSERT INTO customer VALUES(8, 'Maria', '2019-01-08', 80); |
|  | INSERT INTO customer VALUES(9, 'Jaze', '2019-01-09', 110); |
|  | INSERT INTO customer VALUES(1, 'Jhon', '2019-01-10', 130); |
|  | INSERT INTO customer VALUES(3, 'Jade', '2019-01-10', 150); |
|  |  |
|  | WITH daily\_sales AS( |
|  | SELECT |
|  | visited\_on, |
|  | sum(amount) AS daily\_sales\_total\_amount |
|  | FROM |
|  | customer |
|  | GROUP BY visited\_on |
|  | ) |
|  | SELECT |
|  | visited\_on, |
|  | sum(daily\_sales\_total\_amount) |
|  | OVER(ORDER BY visited\_on RANGE BETWEEN INTERVAL '6' day PRECEDING AND CURRENT ROW) AS amount, |
|  | ROUND(avg(daily\_sales\_total\_amount) |
|  | OVER(ORDER BY visited\_on RANGE BETWEEN INTERVAL '6' day PRECEDING AND CURRENT ROW), 2) AS average\_amount |
|  | FROM |
|  | daily\_sales |
|  | LIMIT 6,18446744073709551615 |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE customer; |
|  |  |
|  |  |
|  | --Q68: |
|  |  |
|  | CREATE TABLE scores |
|  | ( |
|  | player\_name VARCHAR(25), |
|  | gender VARCHAR(1), |
|  | day DATE, |
|  | score\_points INT, |
|  | CONSTRAINT pk\_scores PRIMARY KEY (gender, day) |
|  | ); |
|  |  |
|  | INSERT INTO scores VALUES('Aron', 'F', '2020-01-01', 17); |
|  | INSERT INTO scores VALUES('Alice', 'F', '2020-01-07', 23); |
|  | INSERT INTO scores VALUES('Bajrang', 'M', '2020-01-07', 7); |
|  | INSERT INTO scores VALUES('Khali' , 'M', '2019-12-25', 11); |
|  | INSERT INTO scores VALUES('Slaman', 'M', '2019-12-30', 13); |
|  | INSERT INTO scores VALUES('Joe', 'M', '2019-12-31', 3); |
|  | INSERT INTO scores VALUES('Jose', 'M', '2019-12-18', 2); |
|  | INSERT INTO scores VALUES('Priya', 'F', '2019-12-31', 23); |
|  | INSERT INTO scores VALUES('Priyanka', 'F', '2019-12-30', 17); |
|  |  |
|  | SELECT |
|  | gender, |
|  | day, |
|  | sum(score\_points) OVER(PARTITION BY gender ORDER BY day) AS total |
|  | FROM |
|  | scores |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE scores; |
|  |  |
|  |  |
|  | --Q69: |
|  |  |
|  | CREATE TABLE logs |
|  | ( |
|  | log\_id INT, |
|  | CONSTRAINT pk\_logs PRIMARY KEY (log\_id) |
|  | ); |
|  |  |
|  | INSERT INTO logs VALUES(1),(2),(3),(7),(8),(10); |
|  |  |
|  | WITH log\_id\_with\_diff\_with\_next\_val AS ( |
|  | SELECT |
|  | log\_id, |
|  | IFNULL(log\_id - LEAD(log\_id, 1) OVER(ORDER BY log\_id), 0) AS diff\_next\_val, |
|  | IFNULL(log\_id - LAG(log\_id, 1) OVER(ORDER BY log\_id), 0) AS diff\_prev\_val |
|  | FROM |
|  | logs |
|  | ), |
|  | start\_end\_logs AS( |
|  | SELECT |
|  | CASE |
|  | WHEN ABS(diff\_next\_val) = 1 |
|  | OR (ABS(diff\_next\_val) <> 1 |
|  | AND ABS(diff\_prev\_val) <> 1) |
|  | THEN log\_id |
|  | END start\_id, |
|  | CASE |
|  | WHEN ABS(LEAD(diff\_prev\_val) OVER(ORDER BY log\_id)) = 1 |
|  | THEN LEAD(log\_id) OVER(ORDER BY log\_id) |
|  | WHEN (ABS(diff\_next\_val) <> 1 |
|  | AND ABS(diff\_prev\_val) <> 1) |
|  | THEN log\_id |
|  | END end\_id |
|  |  |
|  | FROM |
|  | log\_id\_with\_diff\_with\_next\_val |
|  | WHERE |
|  | NOT ( |
|  | ABS(IFNULL(diff\_next\_val, 0)) = 1 |
|  | AND ABS(IFNULL(diff\_prev\_val, 0)) = 1 |
|  | ) |
|  | ) |
|  | SELECT |
|  | \* |
|  | FROM |
|  | start\_end\_logs |
|  | WHERE |
|  | start\_id IS NOT NULL |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE logs; |
|  |  |
|  |  |
|  | --Q70: |
|  |  |
|  | CREATE TABLE students |
|  | ( |
|  | student\_id INT, |
|  | student\_name VARCHAR(25), |
|  | CONSTRAINT pk\_students PRIMARY KEY (student\_id) |
|  | ); |
|  |  |
|  | CREATE TABLE subjects |
|  | ( |
|  | subject\_name VARCHAR(25), |
|  | CONSTRAINT pk\_students PRIMARY KEY (subject\_name) |
|  | ); |
|  |  |
|  | CREATE TABLE examinations |
|  | ( |
|  | student\_id INT, |
|  | subject\_name VARCHAR(25) |
|  | ); |
|  |  |
|  | INSERT INTO students VALUES(1, 'Alice'); |
|  | INSERT INTO students VALUES(2, 'Bob'); |
|  | INSERT INTO students VALUES(13, 'John'); |
|  | INSERT INTO students VALUES(6,'Alex'); |
|  |  |
|  | INSERT INTO subjects VALUES('Math'); |
|  | INSERT INTO subjects VALUES('Physics'); |
|  | INSERT INTO subjects VALUES('Programming'); |
|  |  |
|  | INSERT INTO examinations VALUES(1,'Math'); |
|  | INSERT INTO examinations VALUES(1, 'Physics'); |
|  | INSERT INTO examinations VALUES(1, 'Programming'); |
|  | INSERT INTO examinations VALUES(2, 'Programming'); |
|  | INSERT INTO examinations VALUES(1, 'Physics'); |
|  | INSERT INTO examinations VALUES(1, 'Math'); |
|  | INSERT INTO examinations VALUES(13, 'Math'); |
|  | INSERT INTO examinations VALUES(13, 'Programming'); |
|  | INSERT INTO examinations VALUES(13, 'Physics'); |
|  | INSERT INTO examinations VALUES(2, 'Math'); |
|  | INSERT INTO examinations VALUES(1, 'Math'); |
|  |  |
|  |  |
|  | SELECT |
|  | st.student\_id, |
|  | st.student\_name, |
|  | sb.subject\_name, |
|  | sum( |
|  | CASE |
|  | WHEN ex.subject\_name IS NOT NULL |
|  | THEN 1 |
|  | ELSE |
|  | 0 |
|  | END |
|  | ) as attended\_exams |
|  | FROM students st |
|  | JOIN subjects sb |
|  | LEFT JOIN examinations ex ON ex.subject\_name = sb.subject\_name |
|  | AND st.student\_id = ex.student\_id |
|  | GROUP BY |
|  | st.student\_id, |
|  | st.student\_name, |
|  | sb.subject\_name |
|  | ORDER BY |
|  | st.student\_id, |
|  | st.student\_name |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE students; |
|  | DROP TABLE subjects; |
|  | DROP TABLE examinations; |
|  |  |
|  |  |
|  | --Q71: |
|  |  |
|  | CREATE TABLE employees |
|  | ( |
|  | employee\_id INT, |
|  | employee\_name VARCHAR(25), |
|  | manager\_id INT, |
|  | CONSTRAINT pk\_employees PRIMARY KEY(employee\_id) |
|  | ); |
|  |  |
|  | INSERT INTO employees VALUES(1, 'Boss', 1); |
|  | INSERT INTO employees VALUES(3, 'Alice', 3); |
|  | INSERT INTO employees VALUES(2, 'Bob', 1); |
|  | INSERT INTO employees VALUES(4, 'Daniel', 2); |
|  | INSERT INTO employees VALUES(7, 'Luis', 4); |
|  | INSERT INTO employees VALUES(8, 'Jhon', 3); |
|  | INSERT INTO employees VALUES(9, 'Angela', 8); |
|  | INSERT INTO employees VALUES(77, 'Robert', 1); |
|  |  |
|  |  |
|  | WITH RECURSIVE emp\_hir AS |
|  | ( |
|  | SELECT |
|  | employee\_id, |
|  | manager\_id, |
|  | employee\_name, |
|  | 1 as lvl |
|  | FROM |
|  | employees |
|  | WHERE |
|  | employee\_name = 'Boss' |
|  | UNION |
|  | SELECT |
|  | em.employee\_id, |
|  | em.manager\_id, |
|  | em.employee\_name, |
|  | eh.lvl + 1 as lvl |
|  | FROM |
|  | emp\_hir eh |
|  | JOIN employees em ON eh.employee\_id = em.manager\_id |
|  | WHERE |
|  | em.employee\_name <> 'Boss' |
|  | ) |
|  | SELECT |
|  | eh1.employee\_id |
|  | FROM |
|  | emp\_hir eh1 |
|  | WHERE |
|  | eh1.employee\_name <> 'Boss' |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE employees; |
|  |  |
|  |  |
|  | --Q72: |
|  |  |
|  | CREATE TABLE transactions |
|  | ( |
|  | id INT, |
|  | country VARCHAR(25), |
|  | state VARCHAR(15), |
|  | amount INT, |
|  | trans\_date DATE, |
|  | CONSTRAINT pk\_trx PRIMARY KEY(id) |
|  | ); |
|  |  |
|  | INSERT INTO transactions VALUES(121, 'US', 'approved', 1000, '2018-12-18'); |
|  | INSERT INTO transactions VALUES(122, 'US', 'declined', 2000, '2018-12-19'); |
|  | INSERT INTO transactions VALUES(123, 'US', 'approved', 2000, '2019-01-01'); |
|  | INSERT INTO transactions VALUES(124, 'DE', 'approved', 2000, '2019-01-07'); |
|  |  |
|  | SELECT |
|  | DATE\_FORMAT(trans\_date, '%Y-%m') AS month, |
|  | country, |
|  | COUNT(\*) AS trans\_count, |
|  | COUNT( |
|  | CASE |
|  | WHEN state = 'approved' |
|  | THEN id |
|  | END |
|  | ) AS approved\_count, |
|  | SUM(amount) AS trans\_total\_amount, |
|  | SUM( |
|  | CASE |
|  | WHEN state = 'approved' |
|  | THEN amount |
|  | END |
|  | ) AS approved\_total\_amount |
|  | FROM |
|  | transactions |
|  | GROUP BY |
|  | DATE\_FORMAT(trans\_date, '%Y-%m'), |
|  | country |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE transactions; |
|  |  |
|  | --Q73: |
|  |  |
|  | CREATE TABLE actions |
|  | ( |
|  | user\_id INT, |
|  | post\_id INT, |
|  | action\_date DATE, |
|  | action VARCHAR(15), |
|  | extra VARCHAR(25) |
|  | ); |
|  |  |
|  | CREATE TABLE removals |
|  | ( |
|  | post\_id INT, |
|  | remove\_date DATE |
|  | ); |
|  |  |
|  | INSERT INTO actions VALUES(1, 1, '2019-07-01','view', null); |
|  | INSERT INTO actions VALUES(1, 1, '2019-07-01', 'like', null); |
|  | INSERT INTO actions VALUES(1, 1, '2019-07-01', 'share', null); |
|  | INSERT INTO actions VALUES(2, 2, '2019-07-04', 'view', null); |
|  | INSERT INTO actions VALUES(2, 2, '2019-07-04', 'report', 'spam'); |
|  | INSERT INTO actions VALUES(3, 4, '2019-07-04', 'view', null); |
|  | INSERT INTO actions VALUES(3, 4, '2019-07-04', 'report', 'spam'); |
|  | INSERT INTO actions VALUES(4, 3, '2019-07-02', 'view', null); |
|  | INSERT INTO actions VALUES(4, 3, '2019-07-02', 'report', 'spam'); |
|  | INSERT INTO actions VALUES(5, 2, '2019-07-03', 'view', null); |
|  | INSERT INTO actions VALUES(5, 2, '2019-07-03', 'report', 'racism'); |
|  | INSERT INTO actions VALUES(5, 5, '2019-07-03', 'view', null); |
|  | INSERT INTO actions VALUES(5, 5, '2019-07-03', 'report', 'racism'); |
|  |  |
|  | INSERT INTO removals VALUES(2, '2019-07-20'); |
|  | INSERT INTO removals VALUES(3, '2019-07-18'); |
|  |  |
|  | WITH reported\_removed\_stat AS( |
|  | SELECT |
|  | a.action\_date, |
|  | COUNT(\*) AS total\_reported, |
|  | COUNT( |
|  | CASE |
|  | WHEN r.post\_id IS NOT NULL |
|  | THEN r.post\_id |
|  | END |
|  | ) AS total\_removed |
|  | FROM |
|  | actions a |
|  | LEFT JOIN removals r ON r.post\_id = a.post\_id |
|  | WHERE |
|  | a.action = 'report' |
|  | AND a.extra = 'spam' |
|  | GROUP BY |
|  | a.action\_date |
|  | ), |
|  | daily\_pct AS( |
|  | SELECT |
|  | action\_date, |
|  | (total\_removed)\*100.00/total\_reported as daily\_ratio |
|  | FROM |
|  | reported\_removed\_stat |
|  | GROUP BY |
|  | action\_date |
|  | ) |
|  | SELECT |
|  | ROUND(AVG(daily\_ratio), 2) AS average\_daily\_percent |
|  | FROM |
|  | daily\_pct |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE actions; |
|  | DROP TABLE removals; |
|  |  |
|  |  |
|  | --Q74: |
|  |  |
|  | CREATE TABLE activity |
|  | ( |
|  | player\_id INT, |
|  | device\_id INT, |
|  | event\_date DATE, |
|  | games\_played INT, |
|  | CONSTRAINT pk\_activity PRIMARY KEY(player\_id, event\_date) |
|  | ); |
|  |  |
|  | INSERT INTO activity VALUES(1, 2, '2016-03-01', 5); |
|  | INSERT INTO activity VALUES(1 ,2, '2016-03-02', 6); |
|  | INSERT INTO activity VALUES(2, 3, '2017-06-25', 1); |
|  | INSERT INTO activity VALUES(3, 1, '2016-03-02', 0); |
|  | INSERT INTO activity VALUES(3, 4, '2018-07-03', 5); |
|  |  |
|  | WITH logged\_in\_prev\_day AS( |
|  | SELECT |
|  | count(DISTINCT player\_id) AS player\_count |
|  | FROM |
|  | activity a1 |
|  | WHERE |
|  | EXISTS( |
|  | SELECT |
|  | \* |
|  | FROM |
|  | activity a2 |
|  | WHERE |
|  | a1.player\_id = a2.player\_id |
|  | AND a1.event\_date = DATE\_ADD(a2.event\_date, INTERVAL 1 DAY) |
|  | ) |
|  | ), |
|  | all\_player AS( |
|  | SELECT |
|  | count(DISTINCT player\_id) AS total\_player\_count |
|  | FROM |
|  | activity |
|  | ) |
|  | SELECT |
|  | ROUND( |
|  | (SELECT |
|  | player\_count |
|  | FROM |
|  | logged\_in\_prev\_day)\*1.00/ |
|  | (SELECT |
|  | total\_player\_count |
|  | FROM |
|  | all\_player) |
|  | ,2) AS fraction |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE activity; |
|  |  |
|  |  |
|  | --Q75: |
|  |  |
|  | CREATE TABLE activity |
|  | ( |
|  | player\_id INT, |
|  | device\_id INT, |
|  | event\_date DATE, |
|  | games\_played INT, |
|  | CONSTRAINT pk\_activity PRIMARY KEY(player\_id, event\_date) |
|  | ); |
|  |  |
|  | INSERT INTO activity VALUES(1, 2, '2016-03-01', 5); |
|  | INSERT INTO activity VALUES(1 ,2, '2016-03-02', 6); |
|  | INSERT INTO activity VALUES(2, 3, '2017-06-25', 1); |
|  | INSERT INTO activity VALUES(3, 1, '2016-03-02', 0); |
|  | INSERT INTO activity VALUES(3, 4, '2018-07-03', 5); |
|  |  |
|  | WITH logged\_in\_prev\_day AS( |
|  | SELECT |
|  | count(DISTINCT player\_id) AS player\_count |
|  | FROM |
|  | activity a1 |
|  | WHERE |
|  | EXISTS( |
|  | SELECT |
|  | \* |
|  | FROM |
|  | activity a2 |
|  | WHERE |
|  | a1.player\_id = a2.player\_id |
|  | AND a1.event\_date = DATE\_ADD(a2.event\_date, INTERVAL 1 DAY) |
|  | ) |
|  | ), |
|  | all\_player AS( |
|  | SELECT |
|  | count(DISTINCT player\_id) AS total\_player\_count |
|  | FROM |
|  | activity |
|  | ) |
|  | SELECT |
|  | ROUND( |
|  | (SELECT |
|  | player\_count |
|  | FROM |
|  | logged\_in\_prev\_day)\*1.00/ |
|  | (SELECT |
|  | total\_player\_count |
|  | FROM |
|  | all\_player) |
|  | ,2) AS fraction |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE activity; |
|  |  |
|  |  |
|  | --Q76: |
|  |  |
|  | CREATE TABLE salaries |
|  | ( |
|  | company\_id INT, |
|  | employee\_id INT, |
|  | employee\_name VARCHAR(25), |
|  | salary INT, |
|  | CONSTRAINT pk\_salary PRIMARY KEY(company\_id, employee\_id) |
|  | ); |
|  |  |
|  | INSERT INTO salaries VALUES(1, 1, 'Tony', 2000); |
|  | INSERT INTO salaries VALUES(1, 2, 'Pronub', 21300); |
|  | INSERT INTO salaries VALUES(1, 3, 'Tyrrox', 10800); |
|  | INSERT INTO salaries VALUES(2, 1, 'Pam', 300); |
|  | INSERT INTO salaries VALUES(2, 7, 'Bassem', 450); |
|  | INSERT INTO salaries VALUES(2, 9, 'Hermione', 700); |
|  | INSERT INTO salaries VALUES(3, 7, 'Bocaben', 100); |
|  | INSERT INTO salaries VALUES(3, 2, 'Ognjen', 2200); |
|  | INSERT INTO salaries VALUES(3, 13, 'Nyan Cat', 3300); |
|  | INSERT INTO salaries VALUES(3, 15, 'Morning Cat', 7777); |
|  |  |
|  | WITH salaries\_with\_max\_by\_company AS ( |
|  | SELECT |
|  | \*, |
|  | MAX(salary) OVER(PARTITION BY company\_id) AS max\_company\_salary |
|  | FROM |
|  | salaries |
|  | ) |
|  | SELECT |
|  | company\_id, |
|  | employee\_id, |
|  | employee\_name, |
|  | CASE |
|  | WHEN max\_company\_salary > 10000 |
|  | THEN |
|  | CEILING(salary - (salary\*0.49)) |
|  | WHEN max\_company\_salary >= 1000 |
|  | THEN |
|  | CEILING(salary - (salary\*0.24)) |
|  | ELSE |
|  | salary |
|  | END AS salary |
|  | FROM |
|  | salaries\_with\_max\_by\_company |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE salaries; |
|  |  |
|  |  |
|  | --Q77: |
|  |  |
|  | CREATE TABLE variables |
|  | ( |
|  | name VARCHAR(5), |
|  | value INT, |
|  | CONSTRAINT pk\_variables PRIMARY KEY(name) |
|  | ); |
|  |  |
|  | CREATE TABLE expressions |
|  | ( |
|  | left\_operand VARCHAR(5), |
|  | operator ENUM('<', '>', '='), |
|  | right\_operand VARCHAR(5), |
|  | CONSTRAINT pk\_expressions PRIMARY KEY(left\_operand, operator, right\_operand) |
|  | ); |
|  |  |
|  | INSERT INTO variables VALUES('x', 66); |
|  | INSERT INTO variables VALUES('y', 77); |
|  |  |
|  | INSERT INTO expressions VALUES('x', '>', 'y'); |
|  | INSERT INTO expressions VALUES('x', '<', 'y'); |
|  | INSERT INTO expressions VALUES('x', '=', 'y'); |
|  | INSERT INTO expressions VALUES('y', '>', 'x'); |
|  | INSERT INTO expressions VALUES('y', '<', 'x'); |
|  | INSERT INTO expressions VALUES('y', '=', 'x'); |
|  |  |
|  | WITH variables\_with\_val AS( |
|  | SELECT |
|  | e.left\_operand, |
|  | e.right\_operand, |
|  | e.operator, |
|  | MAX(CASE |
|  | WHEN e.left\_operand = v.name |
|  | THEN v.value |
|  | END) as first\_var, |
|  | MAX(CASE |
|  | WHEN e.right\_operand = v.name |
|  | THEN v.value |
|  | END) as second\_var |
|  | FROM |
|  | expressions e |
|  | JOIN variables v |
|  | ON e.left\_operand = v.name |
|  | OR e.right\_operand = v.name |
|  | GROUP BY |
|  | left\_operand, |
|  | right\_operand, |
|  | operator |
|  | ) |
|  | SELECT |
|  | left\_operand, |
|  | operator, |
|  | right\_operand, |
|  | CASE |
|  | WHEN operator = '<' |
|  | THEN |
|  | IF(first\_var < second\_var, 'true', 'false') |
|  | WHEN operator = '>' |
|  | THEN |
|  | IF(first\_var > second\_var, 'true', 'false') |
|  | WHEN operator = '=' |
|  | THEN |
|  | IF(first\_var = second\_var, 'true', 'false') |
|  | END value |
|  | FROM |
|  | variables\_with\_val |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE variables; |
|  |  |
|  |  |
|  | --Q78: |
|  |  |
|  | CREATE TABLE person |
|  | ( |
|  | id INT, |
|  | name VARCHAR(25), |
|  | phone\_number VARCHAR(11), |
|  | CONSTRAINT pk\_person PRIMARY KEY (id) |
|  | ); |
|  |  |
|  | CREATE TABLE country |
|  | ( |
|  | name VARCHAR(25), |
|  | country\_code VARCHAR(3), |
|  | CONSTRAINT pk\_country PRIMARY KEY (country\_code) |
|  | ); |
|  |  |
|  | CREATE TABLE calls |
|  | ( |
|  | caller\_id INT, |
|  | callee\_id INT, |
|  | duration INT |
|  | ); |
|  |  |
|  | INSERT INTO person VALUES(3, 'Jonathan', '051-1234567'); |
|  | INSERT INTO person VALUES(12, 'Elvis', '051-7654321'); |
|  | INSERT INTO person VALUES(1, 'Moncef', '212-1234567'); |
|  | INSERT INTO person VALUES(2, 'Maroua', '212-6523651'); |
|  | INSERT INTO person VALUES(7, 'Meir', '972-1234567'); |
|  | INSERT INTO person VALUES(9, 'Rachel', '972-0011100'); |
|  |  |
|  | INSERT INTO country VALUES('Peru', '51'); |
|  | INSERT INTO country VALUES('Israel', '972'); |
|  | INSERT INTO country VALUES('Morocco', '212'); |
|  | INSERT INTO country VALUES('Germany', '49'); |
|  | INSERT INTO country VALUES('Ethiopia', '251'); |
|  |  |
|  | INSERT INTO calls VALUES(1, 9, 33); |
|  | INSERT INTO calls VALUES(2, 9, 4); |
|  | INSERT INTO calls VALUES(1, 2, 59); |
|  | INSERT INTO calls VALUES(3, 12, 102); |
|  | INSERT INTO calls VALUES(3, 12, 330); |
|  | INSERT INTO calls VALUES(12, 3, 5); |
|  | INSERT INTO calls VALUES(7, 9, 13); |
|  | INSERT INTO calls VALUES(7, 1, 3); |
|  | INSERT INTO calls VALUES(9, 7, 1); |
|  | INSERT INTO calls VALUES(1, 7, 7); |
|  |  |
|  | WITH receiver\_caller\_calls AS( |
|  | SELECT |
|  | caller\_id AS caller\_receiver\_id, |
|  | duration |
|  | FROM |
|  | calls |
|  | UNION ALL |
|  | SELECT |
|  | callee\_id AS caller\_receiver\_id, |
|  | duration |
|  | FROM |
|  | calls |
|  | ), |
|  | call\_duration\_avg AS( |
|  | SELECT |
|  | DISTINCT cn.name, |
|  | avg(c.duration) OVER() as global\_average, |
|  | avg(c.duration) OVER(PARTITION BY cn.name) as country\_average |
|  | FROM |
|  | person p |
|  | JOIN country cn |
|  | ON CAST(SUBSTRING\_INDEX(p.phone\_number, '-', 1) AS UNSIGNED) = CAST(cn.country\_code AS UNSIGNED) |
|  | JOIN receiver\_caller\_calls c |
|  | ON c.caller\_receiver\_id = p.id |
|  | ) |
|  | SELECT |
|  | name |
|  | FROM |
|  | call\_duration\_avg |
|  | WHERE |
|  | country\_average > global\_average |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE person; |
|  | DROP TABLE country; |
|  | DROP TABLE calls; |
|  |  |
|  |  |
|  | --Q79: |
|  |  |
|  | CREATE TABLE employee |
|  | ( |
|  | employee\_id INT, |
|  | name VARCHAR(25), |
|  | months INT, |
|  | salary INT, |
|  | CONSTRAINT pk\_employee PRIMARY KEY(employee\_id) |
|  | ); |
|  |  |
|  |  |
|  | INSERT INTO employee VALUES(12228, 'Rose', 15, 1968); |
|  | INSERT INTO employee VALUES(33645, 'Angela', 1, 3443); |
|  | INSERT INTO employee VALUES(45692, 'Frank', 17, 1608); |
|  | INSERT INTO employee VALUES(56118, 'Patrick', 7, 1345); |
|  | INSERT INTO employee VALUES(59725, 'Lisa', 11, 2330); |
|  | INSERT INTO employee VALUES(74197, 'Kimberly', 16, 4372); |
|  | INSERT INTO employee VALUES(78454, 'Bonnie', 8, 1771); |
|  | INSERT INTO employee VALUES(83565, 'Michael', 6, 2017); |
|  | INSERT INTO employee VALUES(98607, 'Todd', 5, 3396); |
|  | INSERT INTO employee VALUES(99989, 'Joe', 9, 3573); |
|  |  |
|  | SELECT |
|  | name |
|  | FROM |
|  | employee |
|  | ORDER BY |
|  | name |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE employee; |
|  |  |
|  |  |
|  | --Q80: |
|  |  |
|  | CREATE TABLE user\_transactions |
|  | ( |
|  | transaction\_id INT, |
|  | product\_id INT, |
|  | spend DECIMAL(10,2), |
|  | transaction\_date DATE |
|  | ); |
|  |  |
|  |  |
|  | INSERT INTO user\_transactions VALUES(1341, 123424, 1500.60, '2019-12-31'); |
|  | INSERT INTO user\_transactions VALUES(1423, 123424, 1000.20, '2020-12-31'); |
|  | INSERT INTO user\_transactions VALUES(1623, 123424, 1246.44, '2021-12-31'); |
|  | INSERT INTO user\_transactions VALUES(1322, 123424, 2145.32, '2022-12-31'); |
|  |  |
|  |  |
|  | SELECT |
|  | DATE\_FORMAT(transaction\_date,'%Y') AS year, |
|  | product\_id, |
|  | spend AS curr\_year\_spend, |
|  | LAG(spend) OVER(ORDER BY DATE\_FORMAT(transaction\_date,'%Y')) AS prev\_year\_spend, |
|  | ROUND((spend - LAG(spend) OVER(ORDER BY DATE\_FORMAT(transaction\_date,'%Y'))) |
|  | \*100.00/LAG(spend) OVER(ORDER BY DATE\_FORMAT(transaction\_date,'%Y')),2) AS yoy\_rate |
|  | FROM |
|  | user\_transactions |
|  | ORDER BY |
|  | year |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE user\_transactions; |
|  |  |
|  |  |
|  | --Q81: |
|  |  |
|  | CREATE TABLE inventory |
|  | ( |
|  | item\_id INT, |
|  | item\_type VARCHAR(20), |
|  | item\_category VARCHAR(20), |
|  | square\_footage DECIMAL(10,2) |
|  | ); |
|  |  |
|  | INSERT INTO inventory VALUES(1374, 'prime\_eligible' , 'mini refrigerator', 68.00); |
|  | INSERT INTO inventory VALUES(4245, 'not\_prime', 'standing lamp', 26.40); |
|  | INSERT INTO inventory VALUES(2452, 'prime\_eligible', 'television', 85.00); |
|  | INSERT INTO inventory VALUES(3255, 'not\_prime', 'side table', 22.60); |
|  | INSERT INTO inventory VALUES(1672, 'prime\_eligible', 'laptop', 8.50); |
|  |  |
|  | WITH product\_inventory\_summary AS |
|  | ( |
|  | SELECT |
|  | item\_type, |
|  | SUM(square\_footage) as square\_footage\_required, |
|  | COUNT(item\_id) as unique\_item\_count, |
|  | 500000 as total\_space, |
|  | FLOOR(500000/sum(square\_footage))\*sum(square\_footage) as space\_used, |
|  | FLOOR(500000/sum(square\_footage))\*COUNT(item\_id) as item\_count |
|  | FROM |
|  | inventory |
|  | GROUP BY |
|  | item\_type |
|  | ) |
|  | SELECT |
|  | t1.item\_type, |
|  | CASE |
|  | WHEN t1.item\_type = 'prime\_eligible' |
|  | THEN t1.item\_count |
|  | ELSE |
|  | FLOOR((500000-t2.space\_used)/t1.square\_footage\_required)\*t1.unique\_item\_count |
|  | END AS item\_count |
|  | FROM |
|  | product\_inventory\_summary t1 |
|  | JOIN product\_inventory\_summary t2 ON t1.item\_type <> t2.item\_type |
|  | ORDER BY t1.item\_type DESC |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE inventory; |
|  |  |
|  |  |
|  | --Q82: |
|  |  |
|  | CREATE TABLE user\_actions |
|  | ( |
|  | user\_id INT, |
|  | event\_id INT, |
|  | event\_type VARCHAR(20), |
|  | event\_date DATE |
|  | ); |
|  |  |
|  | INSERT INTO user\_actions VALUES(445, 7765 , 'sign-in', '2022-05-31'); |
|  | INSERT INTO user\_actions VALUES(742, 6458, 'sign-in', '2022-06-03'); |
|  | INSERT INTO user\_actions VALUES(445, 3634, 'like', '2022-06-05'); |
|  | INSERT INTO user\_actions VALUES(742, 1374, 'comment', '2022-06-05'); |
|  | INSERT INTO user\_actions VALUES(648, 3124, 'like', '2022-06-18'); |
|  |  |
|  |  |
|  | SELECT |
|  | CAST(DATE\_FORMAT(curr\_month\_ua.event\_date, '%m') AS UNSIGNED) AS month, |
|  | count(distinct curr\_month\_ua.user\_id) AS monthly\_active\_users |
|  | FROM |
|  | user\_actions curr\_month\_ua |
|  | WHERE |
|  | curr\_month\_ua.event\_type IN ('sign-in', 'like', 'comment') |
|  | AND DATE\_FORMAT(curr\_month\_ua.event\_date,'%Y-%m') = '2022-06' |
|  | AND EXISTS( |
|  | SELECT |
|  | \* |
|  | FROM |
|  | user\_actions last\_month\_ua |
|  | WHERE |
|  | curr\_month\_ua.user\_id = last\_month\_ua.user\_id |
|  | AND last\_month\_ua.event\_type IN ('sign-in', 'like', 'comment') |
|  | AND DATE\_FORMAT(curr\_month\_ua.event\_date, '%Y-%m') = |
|  | DATE\_FORMAT(last\_month\_ua.event\_date + INTERVAL '1' MONTH, '%Y-%m') |
|  | ) |
|  | GROUP BY |
|  | CAST(DATE\_FORMAT(curr\_month\_ua.event\_date, '%m') AS UNSIGNED) |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE user\_actions; |
|  |  |
|  |  |
|  | --Q83: |
|  |  |
|  | CREATE TABLE search\_frequency |
|  | ( |
|  | searches INT, |
|  | num\_users INT |
|  | ); |
|  |  |
|  | INSERT INTO search\_frequency VALUES(1, 2); |
|  | INSERT INTO search\_frequency VALUES(2, 2); |
|  | INSERT INTO search\_frequency VALUES(3, 3); |
|  | INSERT INTO search\_frequency VALUES(4, 1); |
|  |  |
|  | WITH cumulative\_sum AS |
|  | ( |
|  | SELECT |
|  | \*, |
|  | SUM(num\_users) OVER(ORDER BY searches) as cum\_sum, |
|  | ROW\_NUMBER() OVER(ORDER BY searches) as row\_num |
|  | FROM |
|  | search\_frequency |
|  | ), |
|  | max\_cumulative\_sum AS |
|  | ( |
|  | SELECT |
|  | MAX(cum\_sum) as max\_cum\_sum |
|  | FROM |
|  | cumulative\_sum |
|  | ), |
|  | odd\_even\_sum AS |
|  | ( |
|  | SELECT |
|  | MIN(row\_num) AS row1 |
|  | FROM |
|  | cumulative\_sum |
|  | WHERE |
|  | cum\_sum >= ( |
|  | SELECT |
|  | CEILING((max\_cum\_sum)\*0.5) |
|  | FROM |
|  | max\_cumulative\_sum |
|  | ) |
|  | ), |
|  | even\_sum AS |
|  | ( |
|  | SELECT |
|  | MIN(row\_num) AS row2 |
|  | FROM |
|  | cumulative\_sum |
|  | WHERE |
|  | cum\_sum >= ( |
|  | SELECT |
|  | CEILING((max\_cum\_sum)\*0.5)+1 |
|  | FROM |
|  | max\_cumulative\_sum |
|  | ) |
|  | ) |
|  | SELECT |
|  | ROUND( |
|  | CASE |
|  | WHEN (SELECT max\_cum\_sum FROM max\_cumulative\_sum) % 2 = 0 |
|  | THEN ( |
|  | (SELECT searches FROM cumulative\_sum WHERE row\_num = (SELECT row1 FROM odd\_even\_sum)) |
|  | + |
|  | (SELECT searches FROM cumulative\_sum WHERE row\_num = (SELECT row2 FROM even\_sum)) |
|  | )/2.0 |
|  | ELSE (SELECT searches FROM cumulative\_sum WHERE row\_num = (SELECT row1 FROM odd\_even\_sum)) |
|  | END |
|  | ,1) as median |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE search\_frequency; |
|  |  |
|  |  |
|  | --Q84: |
|  |  |
|  | CREATE TABLE advertiser |
|  | ( |
|  | user\_id VARCHAR(15), |
|  | status VARCHAR(15) |
|  | ); |
|  |  |
|  | CREATE TABLE daily\_pay |
|  | ( |
|  | user\_id VARCHAR(15), |
|  | paid DECIMAL(10,2) |
|  | ); |
|  |  |
|  | INSERT INTO advertiser VALUES('bing', 'NEW'); |
|  | INSERT INTO advertiser VALUES('yahoo', 'NEW'); |
|  | INSERT INTO advertiser VALUES('alibaba', 'EXISTING'); |
|  |  |
|  | --resuccernt test |
|  | INSERT INTO advertiser VALUES('oracle', 'CHURN'); |
|  |  |
|  | INSERT INTO daily\_pay VALUES('yahoo', '45.00'); |
|  | INSERT INTO daily\_pay VALUES('alibaba', '100.00'); |
|  | INSERT INTO daily\_pay VALUES('target', '13.00'); |
|  |  |
|  | --resuccernt test |
|  | INSERT INTO daily\_pay VALUES('oracle', '13.00'); |
|  |  |
|  |  |
|  |  |
|  | WITH full\_outer\_join\_table AS( |
|  | SELECT |
|  | dp.user\_id AS user\_id\_dp, |
|  | dp.paid, |
|  | a.user\_id AS user\_id\_advertiser, |
|  | a.status |
|  | FROM daily\_pay dp |
|  | LEFT OUTER JOIN advertiser a ON dp.user\_id = a.user\_id |
|  | UNION |
|  | SELECT |
|  | dp.user\_id AS user\_id\_dp, |
|  | dp.paid, |
|  | a.user\_id AS user\_id\_advertiser, |
|  | a.status |
|  | FROM daily\_pay dp |
|  | RIGHT OUTER JOIN advertiser a ON dp.user\_id = a.user\_id |
|  | ) |
|  | SELECT |
|  | CASE |
|  | WHEN user\_id\_dp IS NULL |
|  | THEN user\_id\_advertiser |
|  | ELSE user\_id\_dp |
|  | END AS user\_id, |
|  | CASE |
|  | WHEN user\_id\_dp is NULL |
|  | THEN |
|  | 'CHURN' |
|  | ELSE |
|  | CASE |
|  | WHEN status is null |
|  | THEN 'NEW' |
|  | WHEN status = 'CHURN' |
|  | THEN 'RESURRECT' |
|  | ELSE 'EXISTING' |
|  | END |
|  | END AS new\_status |
|  | FROM full\_outer\_join\_table full\_table |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE advertiser; |
|  | DROP TABLE daily\_pay; |
|  |  |
|  |  |
|  | --Q85: |
|  |  |
|  | CREATE TABLE server\_utilization |
|  | ( |
|  | server\_id INT, |
|  | status\_time TIMESTAMP, |
|  | session\_status VARCHAR(10) |
|  | ); |
|  |  |
|  | INSERT INTO server\_utilization VALUES(1, '2022-08-02 10:00:00', 'start'); |
|  | INSERT INTO server\_utilization VALUES(1, '2022-08-04 10:00:00', 'stop'); |
|  | INSERT INTO server\_utilization VALUES(2, '2022-08-17 10:00:00', 'start'); |
|  | INSERT INTO server\_utilization VALUES(2, '2022-08-24 10:00:00', 'stop'); |
|  |  |
|  | WITH up\_time\_by\_server AS |
|  | ( |
|  | SELECT |
|  | server\_id, |
|  | session\_status, |
|  | status\_time, |
|  | CASE |
|  | WHEN session\_status = 'stop' |
|  | THEN |
|  | TIMESTAMPDIFF(SECOND, LAG(status\_time) OVER(PARTITION BY server\_id ORDER BY status\_time), status\_time)/3600 |
|  | END as up\_time |
|  | FROM server\_utilization |
|  | ) |
|  | SELECT |
|  | ROUND(sum(up\_time)/24) |
|  | FROM |
|  | up\_time\_by\_server |
|  | WHERE |
|  | up\_time is not null |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE server\_utilization; |
|  |  |
|  |  |
|  | --Q86: |
|  |  |
|  | CREATE TABLE transactions |
|  | ( |
|  | transaction\_id INT, |
|  | merchant\_id INT, |
|  | credit\_card\_id INT, |
|  | amount INT, |
|  | transaction\_timestamp TIMESTAMP |
|  | ); |
|  |  |
|  | INSERT INTO transactions VALUES(1, 101, 1, 100, '2022-09-25 12:00:00'); |
|  | INSERT INTO transactions VALUES(2, 101, 1, 100, '2022-09-25 12:08:00'); |
|  | INSERT INTO transactions VALUES(3, 101, 1, 100, '2022-09-25 12:28:00'); |
|  | INSERT INTO transactions VALUES(4, 102, 2, 300, '2022-09-25 12:00:00'); |
|  | INSERT INTO transactions VALUES(5, 102, 2, 400, '2022-09-25 14:00:00'); |
|  |  |
|  | WITH trx\_with\_repeadted AS |
|  | ( |
|  | SELECT |
|  | credit\_card\_id, |
|  | amount, |
|  | transaction\_timestamp, |
|  | count(\*) OVER( |
|  | PARTITION BY credit\_card\_id,amount |
|  | ORDER BY transaction\_timestamp |
|  | RANGE BETWEEN INTERVAL '10' MINUTE PRECEDING AND CURRENT ROW |
|  | ) AS moving\_count |
|  | FROM |
|  | transactions |
|  | ) |
|  | SELECT |
|  | COUNT(\*) as payment\_count |
|  | FROM trx\_with\_repeadted |
|  | WHERE |
|  | moving\_count > 1 |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE transactions; |
|  |  |
|  |  |
|  | --Q87: |
|  |  |
|  | CREATE TABLE orders |
|  | ( |
|  | order\_id INT, |
|  | customer\_id INT, |
|  | trip\_id INT, |
|  | status VARCHAR(30), |
|  | order\_timestamp TIMESTAMP |
|  | ); |
|  |  |
|  | CREATE TABLE trips |
|  | ( |
|  | dasher\_id INT, |
|  | trip\_id INT, |
|  | estimated\_delivery\_timestamp TIMESTAMP, |
|  | actual\_delivery\_timestamp TIMESTAMP |
|  | ); |
|  |  |
|  | CREATE TABLE customers |
|  | ( |
|  | customer\_id INT, |
|  | signup\_timestamp TIMESTAMP |
|  | ); |
|  |  |
|  | INSERT INTO orders VALUES(727424,8472, 100463, 'completed successfully', '2022-06-05 09:12:00'); |
|  | INSERT INTO orders VALUES(242513, 2341, 100482, 'completed incorrectly', '2022-06-05 14:40:00'); |
|  | INSERT INTO orders VALUES(141367, 1314, 100362, 'completed incorrectly', '2022-06-07 15:03:00'); |
|  | INSERT INTO orders VALUES(582193, 5421, 100657, 'never\_received', '2022-07-07 15:22:00'); |
|  | INSERT INTO orders VALUES(253613, 1314, 100213, 'completed successfully', '2022-06-12 13:43:00'); |
|  |  |
|  | INSERT INTO trips VALUES(101, 100463, '2022-06-05 09:42:00', '2022-06-05 09:38:00'); |
|  | INSERT INTO trips VALUES(102, 100482, '2022-06-05 15:10:00', '2022-06-05 15:46:00'); |
|  | INSERT INTO trips VALUES(101, 100362, '2022-06-07 15:33:00', '2022-06-07 16:45:00'); |
|  | INSERT INTO trips VALUES(102, 100657, '2022-07-07 15:52:00',null); |
|  | INSERT INTO trips VALUES(103, 100213, '2022-06-12 14:13:00', '2022-06-12 14:10:00'); |
|  |  |
|  | INSERT INTO customers VALUES(8472, '2022-05-30 00:00:00'); |
|  | INSERT INTO customers VALUES(2341, '2022-06-01 00:00:00'); |
|  | INSERT INTO customers VALUES(1314, '2022-06-03 00:00:00'); |
|  | INSERT INTO customers VALUES(1435, '2022-06-05 00:00:00'); |
|  | INSERT INTO customers VALUES(5421, '2022-06-07 00:00:00'); |
|  |  |
|  | SELECT |
|  | ROUND((COUNT( |
|  | CASE |
|  | WHEN lower(o.status) <> 'completed successfully' |
|  | THEN o.order\_id |
|  | END |
|  | )\*100.00/COUNT(o.order\_id)),2) as bad\_experience\_pct |
|  |  |
|  | FROM orders o |
|  | JOIN customers c ON c.customer\_id = o.customer\_id |
|  | WHERE |
|  | TIMESTAMPDIFF(DAY,o.order\_timestamp,c.signup\_timestamp) < 14 |
|  | AND DATE\_FORMAT(c.signup\_timestamp,'%Y-%m') = '2022-06' |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE orders; |
|  | DROP TABLE customers; |
|  |  |
|  |  |
|  | --Q88: |
|  |  |
|  | CREATE TABLE scores |
|  | ( |
|  | player\_name VARCHAR(25), |
|  | gender VARCHAR(1), |
|  | day DATE, |
|  | score\_points INT, |
|  | CONSTRAINT pk\_scores PRIMARY KEY (gender, day) |
|  | ); |
|  |  |
|  | INSERT INTO scores VALUES('Aron', 'F', '2020-01-01', 17); |
|  | INSERT INTO scores VALUES('Alice', 'F', '2020-01-07', 23); |
|  | INSERT INTO scores VALUES('Bajrang', 'M', '2020-01-07', 7); |
|  | INSERT INTO scores VALUES('Khali' , 'M', '2019-12-25', 11); |
|  | INSERT INTO scores VALUES('Slaman', 'M', '2019-12-30', 13); |
|  | INSERT INTO scores VALUES('Joe', 'M', '2019-12-31', 3); |
|  | INSERT INTO scores VALUES('Jose', 'M', '2019-12-18', 2); |
|  | INSERT INTO scores VALUES('Priya', 'F', '2019-12-31', 23); |
|  | INSERT INTO scores VALUES('Priyanka', 'F', '2019-12-30', 17); |
|  |  |
|  | SELECT |
|  | gender, |
|  | day, |
|  | sum(score\_points) OVER(PARTITION BY gender ORDER BY day) AS total |
|  | FROM |
|  | scores |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE scores; |
|  |  |
|  |  |
|  | --Q89: |
|  |  |
|  | CREATE TABLE person |
|  | ( |
|  | id INT, |
|  | name VARCHAR(25), |
|  | phone\_number VARCHAR(11), |
|  | CONSTRAINT pk\_person PRIMARY KEY (id) |
|  | ); |
|  |  |
|  | CREATE TABLE country |
|  | ( |
|  | name VARCHAR(25), |
|  | country\_code VARCHAR(3), |
|  | CONSTRAINT pk\_country PRIMARY KEY (country\_code) |
|  | ); |
|  |  |
|  | CREATE TABLE calls |
|  | ( |
|  | caller\_id INT, |
|  | callee\_id INT, |
|  | duration INT |
|  | ); |
|  |  |
|  | INSERT INTO person VALUES(3, 'Jonathan', '051-1234567'); |
|  | INSERT INTO person VALUES(12, 'Elvis', '051-7654321'); |
|  | INSERT INTO person VALUES(1, 'Moncef', '212-1234567'); |
|  | INSERT INTO person VALUES(2, 'Maroua', '212-6523651'); |
|  | INSERT INTO person VALUES(7, 'Meir', '972-1234567'); |
|  | INSERT INTO person VALUES(9, 'Rachel', '972-0011100'); |
|  |  |
|  | INSERT INTO country VALUES('Peru', '51'); |
|  | INSERT INTO country VALUES('Israel', '972'); |
|  | INSERT INTO country VALUES('Morocco', '212'); |
|  | INSERT INTO country VALUES('Germany', '49'); |
|  | INSERT INTO country VALUES('Ethiopia', '251'); |
|  |  |
|  | INSERT INTO calls VALUES(1, 9, 33); |
|  | INSERT INTO calls VALUES(2, 9, 4); |
|  | INSERT INTO calls VALUES(1, 2, 59); |
|  | INSERT INTO calls VALUES(3, 12, 102); |
|  | INSERT INTO calls VALUES(3, 12, 330); |
|  | INSERT INTO calls VALUES(12, 3, 5); |
|  | INSERT INTO calls VALUES(7, 9, 13); |
|  | INSERT INTO calls VALUES(7, 1, 3); |
|  | INSERT INTO calls VALUES(9, 7, 1); |
|  | INSERT INTO calls VALUES(1, 7, 7); |
|  |  |
|  | WITH receiver\_caller\_calls AS( |
|  | SELECT |
|  | caller\_id AS caller\_receiver\_id, |
|  | duration |
|  | FROM |
|  | calls |
|  | UNION ALL |
|  | SELECT |
|  | callee\_id AS caller\_receiver\_id, |
|  | duration |
|  | FROM |
|  | calls |
|  | ), |
|  | call\_duration\_avg AS( |
|  | SELECT |
|  | DISTINCT cn.name, |
|  | avg(c.duration) OVER() as global\_average, |
|  | avg(c.duration) OVER(PARTITION BY cn.name) as country\_average |
|  | FROM |
|  | person p |
|  | JOIN country cn |
|  | ON CAST(SUBSTRING\_INDEX(p.phone\_number, '-', 1) AS UNSIGNED) = CAST(cn.country\_code AS UNSIGNED) |
|  | JOIN receiver\_caller\_calls c |
|  | ON c.caller\_receiver\_id = p.id |
|  | ) |
|  | SELECT |
|  | name |
|  | FROM |
|  | call\_duration\_avg |
|  | WHERE |
|  | country\_average > global\_average |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE person; |
|  | DROP TABLE country; |
|  | DROP TABLE calls; |
|  |  |
|  |  |
|  | --Q90: |
|  |  |
|  | CREATE TABLE numbers |
|  | ( |
|  | num INT, |
|  | frequency INT |
|  | ); |
|  |  |
|  | INSERT INTO numbers VALUES(0, 7); |
|  | INSERT INTO numbers VALUES(1, 1); |
|  | INSERT INTO numbers VALUES(2, 3); |
|  | INSERT INTO numbers VALUES(3, 1); |
|  |  |
|  | WITH cumulative\_sum AS |
|  | ( |
|  | SELECT |
|  | \*, |
|  | SUM(frequency) OVER(ORDER BY num) as cum\_sum, |
|  | ROW\_NUMBER() OVER(ORDER BY num) as row\_num |
|  | FROM |
|  | numbers |
|  | ), |
|  | max\_cumulative\_sum AS |
|  | ( |
|  | SELECT |
|  | MAX(cum\_sum) as max\_cum\_sum |
|  | FROM |
|  | cumulative\_sum |
|  | ), |
|  | odd\_even\_sum AS |
|  | ( |
|  | SELECT |
|  | MIN(row\_num) AS row1 |
|  | FROM |
|  | cumulative\_sum |
|  | WHERE |
|  | cum\_sum >= ( |
|  | SELECT |
|  | CEILING((max\_cum\_sum)\*0.5) |
|  | FROM |
|  | max\_cumulative\_sum |
|  | ) |
|  | ), |
|  | even\_sum AS |
|  | ( |
|  | SELECT |
|  | MIN(row\_num) AS row2 |
|  | FROM |
|  | cumulative\_sum |
|  | WHERE |
|  | cum\_sum >= ( |
|  | SELECT |
|  | CEILING((max\_cum\_sum)\*0.5)+1 |
|  | FROM |
|  | max\_cumulative\_sum |
|  | ) |
|  | ) |
|  | SELECT |
|  | ROUND( |
|  | CASE |
|  | WHEN (SELECT max\_cum\_sum FROM max\_cumulative\_sum) % 2 = 0 |
|  | THEN ( |
|  | (SELECT num FROM cumulative\_sum WHERE row\_num = (SELECT row1 FROM odd\_even\_sum)) |
|  | + |
|  | (SELECT num FROM cumulative\_sum WHERE row\_num = (SELECT row2 FROM even\_sum )) |
|  | )/2.0 |
|  | ELSE( |
|  | SELECT num FROM cumulative\_sum WHERE row\_num = (SELECT row1 FROM odd\_even\_sum) |
|  | ) |
|  | END |
|  | ,1) as median |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE numbers; |
|  |  |
|  |  |
|  | --Q91: |
|  |  |
|  | CREATE TABLE employee |
|  | ( |
|  | employee\_id INT, |
|  | department\_id INT, |
|  | CONSTRAINT pk\_employee PRIMARY KEY(employee\_id) |
|  | ); |
|  |  |
|  | CREATE TABLE salary |
|  | ( |
|  | id INT, |
|  | employee\_id INT, |
|  | amount INT, |
|  | pay\_date DATE, |
|  | CONSTRAINT pk\_salary PRIMARY KEY(id), |
|  | CONSTRAINT fk\_employee FOREIGN KEY(employee\_id) |
|  | REFERENCES employee(employee\_id) |
|  | ); |
|  |  |
|  | INSERT INTO employee VALUES(1, 1); |
|  | INSERT INTO employee VALUES(2, 2); |
|  | INSERT INTO employee VALUES(3, 2); |
|  |  |
|  | INSERT INTO salary VALUES(1, 1, 9000, '2017-03-31'); |
|  | INSERT INTO salary VALUES(2, 2, 6000, '2017-03-31'); |
|  | INSERT INTO salary VALUES(3, 3, 10000, '2017-03-31'); |
|  | INSERT INTO salary VALUES(4, 1, 7000, '2017-02-28'); |
|  | INSERT INTO salary VALUES(5, 2, 6000, '2017-02-28'); |
|  | INSERT INTO salary VALUES(6, 3, 8000, '2017-02-28'); |
|  |  |
|  | WITH department\_company\_avg\_monthly AS( |
|  | SELECT |
|  | DISTINCT DATE\_FORMAT(s.pay\_date, '%Y-%m') AS pay\_month, |
|  | department\_id, |
|  | AVG(amount) OVER(PARTITION BY DATE\_FORMAT(s.pay\_date, '%Y-%m')) as company\_avg, |
|  | AVG(amount) OVER(PARTITION BY DATE\_FORMAT(s.pay\_date, '%Y-%m'), department\_id) as department\_avg |
|  | FROM |
|  | salary s |
|  | JOIN employee e ON s.employee\_id = e.employee\_id |
|  | ) |
|  | SELECT |
|  | pay\_month, |
|  | department\_id, |
|  | CASE |
|  | WHEN department\_avg > company\_avg |
|  | THEN 'higher' |
|  | WHEN department\_avg < company\_avg |
|  | THEN 'lower' |
|  | ELSE |
|  | 'same' |
|  | END AS comparison |
|  | FROM |
|  | department\_company\_avg\_monthly |
|  | ORDER BY |
|  | department\_id |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE salary; |
|  | DROP TABLE employee; |
|  |  |
|  |  |
|  | --Q92: |
|  |  |
|  | CREATE TABLE activity |
|  | ( |
|  | player\_id INT, |
|  | device\_id INT, |
|  | event\_date DATE, |
|  | games\_played INT, |
|  | CONSTRAINT pk\_activity PRIMARY KEY(player\_id, event\_date) |
|  | ); |
|  |  |
|  | INSERT INTO activity VALUES(1, 2, '2016-03-01', 5); |
|  | INSERT INTO activity VALUES(1, 2, '2016-03-02', 6); |
|  | INSERT INTO activity VALUES(2, 3, '2017-06-25', 1); |
|  | INSERT INTO activity VALUES(3, 1, '2016-03-01', 0); |
|  | INSERT INTO activity VALUES(3, 4, '2016-07-03', 5); |
|  |  |
|  | WITH retention\_data AS( |
|  | SELECT |
|  | curr\_day.event\_date, |
|  | COUNT(DISTINCT curr\_day.player\_id) AS retention\_player\_count |
|  | FROM |
|  | activity curr\_day |
|  | WHERE |
|  | EXISTS( |
|  | SELECT |
|  | \* |
|  | FROM |
|  | activity next\_day |
|  | WHERE |
|  | curr\_day.player\_id = next\_day.player\_id |
|  | AND next\_day.event\_date = curr\_day.event\_date + INTERVAL '1' DAY |
|  | ) |
|  | GROUP BY |
|  | curr\_day.event\_date |
|  | ), |
|  | player\_signup\_data AS( |
|  | SELECT |
|  | DISTINCT player\_id, |
|  | FIRST\_VALUE(event\_date) OVER(PARTITION BY player\_id ORDER BY event\_date) as signup\_date |
|  | FROM |
|  | activity |
|  | ), |
|  | daily\_player\_data AS( |
|  | SELECT |
|  | signup\_date, |
|  | COUNT(DISTINCT player\_id) AS daily\_player\_count |
|  | FROM |
|  | player\_signup\_data |
|  | GROUP BY |
|  | signup\_date |
|  | ) |
|  | SELECT |
|  | dpd.signup\_date AS install\_dt, |
|  | dpd.daily\_player\_count AS installs, |
|  | ROUND(IFNULL(rd.retention\_player\_count,0)/dpd.daily\_player\_count,2) AS Day1\_retention |
|  |  |
|  | FROM |
|  | daily\_player\_data dpd |
|  | LEFT JOIN retention\_data rd ON dpd.signup\_date = rd.event\_date |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE activity; |
|  |  |
|  |  |
|  | --Q93: |
|  |  |
|  | CREATE TABLE players |
|  | ( |
|  | player\_id INT, |
|  | group\_id INT, |
|  | CONSTRAINT pk\_players PRIMARY KEY(player\_id) |
|  | ); |
|  |  |
|  | CREATE TABLE matches |
|  | ( |
|  | match\_id INT, |
|  | first\_player INT, |
|  | second\_player INT, |
|  | first\_score INT, |
|  | second\_score INT, |
|  | CONSTRAINT pk\_matches PRIMARY KEY(match\_id) |
|  | ); |
|  |  |
|  | INSERT INTO players VALUES(15, 1); |
|  | INSERT INTO players VALUES(25, 1); |
|  | INSERT INTO players VALUES(30, 1); |
|  | INSERT INTO players VALUES(45, 1); |
|  | INSERT INTO players VALUES(10, 2); |
|  | INSERT INTO players VALUES(35, 2); |
|  | INSERT INTO players VALUES(50, 2); |
|  | INSERT INTO players VALUES(20, 3); |
|  | INSERT INTO players VALUES(40, 3); |
|  |  |
|  | INSERT INTO matches VALUES(1, 15, 45, 3, 0); |
|  | INSERT INTO matches VALUES(2, 30, 25, 1, 2); |
|  | INSERT INTO matches VALUES(3, 30, 15, 2, 0); |
|  | INSERT INTO matches VALUES(4, 40, 20, 5, 2); |
|  | INSERT INTO matches VALUES(5, 35, 50, 1, 1); |
|  |  |
|  | WITH player\_score AS( |
|  | SELECT |
|  | p.group\_id, |
|  | p.player\_id, |
|  | SUM(CASE |
|  | WHEN p.player\_id = m.first\_player |
|  | THEN m.first\_score |
|  | WHEN p.player\_id = m.second\_player |
|  | THEN m.second\_score |
|  | END) AS score |
|  | FROM |
|  | players p |
|  | JOIN matches m ON p.player\_id = m.first\_player OR p.player\_id = m.second\_player |
|  | GROUP BY |
|  | p.group\_id, |
|  | p.player\_id |
|  | ), |
|  | ranked\_player AS( |
|  | SELECT |
|  | group\_id, |
|  | player\_id, |
|  | score, |
|  | DENSE\_RANK() OVER (PARTITION BY group\_id ORDER BY score DESC,player\_id) AS player\_rank |
|  | FROM |
|  | player\_score |
|  | ) |
|  | SELECT |
|  | group\_id, |
|  | player\_id |
|  | FROM |
|  | ranked\_player |
|  | WHERE |
|  | player\_rank = 1 |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE players; |
|  | DROP TABLE matches; |
|  |  |
|  |  |
|  | --Q94: |
|  |  |
|  | CREATE TABLE student |
|  | ( |
|  | student\_id INT, |
|  | student\_name VARCHAR(25), |
|  | CONSTRAINT pk\_student PRIMARY KEY(student\_id) |
|  | ); |
|  |  |
|  | CREATE TABLE exam |
|  | ( |
|  | exam\_id INT, |
|  | student\_id INT, |
|  | score INT, |
|  | CONSTRAINT pk\_exam PRIMARY KEY(exam\_id, student\_id) |
|  | ); |
|  |  |
|  | INSERT INTO student VALUES(1, 'Daniel'); |
|  | INSERT INTO student VALUES(2, 'Jade'); |
|  | INSERT INTO student VALUES(3, 'Stella'); |
|  | INSERT INTO student VALUES(4, 'Jonathan'); |
|  | INSERT INTO student VALUES(5, 'Will'); |
|  |  |
|  | INSERT INTO exam VALUES(10, 1, 70); |
|  | INSERT INTO exam VALUES(10, 2, 80); |
|  | INSERT INTO exam VALUES(10, 3, 90); |
|  | INSERT INTO exam VALUES(20, 1, 80); |
|  | INSERT INTO exam VALUES(30, 1, 70); |
|  | INSERT INTO exam VALUES(30, 3, 80); |
|  | INSERT INTO exam VALUES(30, 4, 90); |
|  | INSERT INTO exam VALUES(40, 1, 60); |
|  | INSERT INTO exam VALUES(40, 2, 70); |
|  | INSERT INTO exam VALUES(40, 4, 80); |
|  |  |
|  | WITH exam\_highest\_lowest AS( |
|  | SELECT |
|  | \*, |
|  | FIRST\_VALUE(score) OVER(PARTITION BY exam\_id ORDER BY score) as exam\_lowest, |
|  | FIRST\_VALUE(score) OVER(PARTITION BY exam\_id ORDER BY score DESC) as exam\_highest |
|  | FROM |
|  | exam |
|  | ), |
|  | student\_highest\_lowest AS( |
|  | SELECT |
|  | DISTINCT student\_id |
|  | FROM |
|  | exam\_highest\_lowest |
|  | WHERE |
|  | score = exam\_lowest |
|  | OR score = exam\_highest |
|  | ) |
|  | SELECT |
|  | student\_id, |
|  | student\_name |
|  | FROM |
|  | student s |
|  | WHERE |
|  | EXISTS( |
|  | SELECT |
|  | \* |
|  | FROM |
|  | exam e |
|  | WHERE |
|  | e.student\_id = s.student\_id |
|  | ) |
|  | AND s.student\_id NOT IN( |
|  | SELECT |
|  | student\_id |
|  | FROM |
|  | student\_highest\_lowest |
|  | ) |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE student; |
|  | DROP TABLE exam; |
|  |  |
|  |  |
|  | --Q95: |
|  |  |
|  | CREATE TABLE student |
|  | ( |
|  | student\_id INT, |
|  | student\_name VARCHAR(25), |
|  | CONSTRAINT pk\_student PRIMARY KEY(student\_id) |
|  | ); |
|  |  |
|  | CREATE TABLE exam |
|  | ( |
|  | exam\_id INT, |
|  | student\_id INT, |
|  | score INT, |
|  | CONSTRAINT pk\_exam PRIMARY KEY(exam\_id, student\_id) |
|  | ); |
|  |  |
|  | INSERT INTO student VALUES(1, 'Daniel'); |
|  | INSERT INTO student VALUES(2, 'Jade'); |
|  | INSERT INTO student VALUES(3, 'Stella'); |
|  | INSERT INTO student VALUES(4, 'Jonathan'); |
|  | INSERT INTO student VALUES(5, 'Will'); |
|  |  |
|  | INSERT INTO exam VALUES(10, 1, 70); |
|  | INSERT INTO exam VALUES(10, 2, 80); |
|  | INSERT INTO exam VALUES(10, 3, 90); |
|  | INSERT INTO exam VALUES(20, 1, 80); |
|  | INSERT INTO exam VALUES(30, 1, 70); |
|  | INSERT INTO exam VALUES(30, 3, 80); |
|  | INSERT INTO exam VALUES(30, 4, 90); |
|  | INSERT INTO exam VALUES(40, 1, 60); |
|  | INSERT INTO exam VALUES(40, 2, 70); |
|  | INSERT INTO exam VALUES(40, 4, 80); |
|  |  |
|  | WITH exam\_highest\_lowest AS( |
|  | SELECT |
|  | \*, |
|  | FIRST\_VALUE(score) OVER(PARTITION BY exam\_id ORDER BY score) as exam\_lowest, |
|  | FIRST\_VALUE(score) OVER(PARTITION BY exam\_id ORDER BY score DESC) as exam\_highest |
|  | FROM |
|  | exam |
|  | ), |
|  | student\_highest\_lowest AS( |
|  | SELECT |
|  | DISTINCT student\_id |
|  | FROM |
|  | exam\_highest\_lowest |
|  | WHERE |
|  | score = exam\_lowest |
|  | OR score = exam\_highest |
|  | ) |
|  | SELECT |
|  | student\_id, |
|  | student\_name |
|  | FROM |
|  | student s |
|  | WHERE |
|  | EXISTS( |
|  | SELECT |
|  | \* |
|  | FROM |
|  | exam e |
|  | WHERE |
|  | e.student\_id = s.student\_id |
|  | ) |
|  | AND s.student\_id NOT IN( |
|  | SELECT |
|  | student\_id |
|  | FROM |
|  | student\_highest\_lowest |
|  | ) |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE student; |
|  | DROP TABLE exam; |
|  |  |
|  |  |
|  | --Q96: |
|  |  |
|  | CREATE TABLE songs\_history |
|  | ( |
|  | history\_id INT, |
|  | user\_id INT, |
|  | song\_id INT, |
|  | song\_plays INT |
|  | ); |
|  |  |
|  | CREATE TABLE songs\_weekly |
|  | ( |
|  | user\_id INT, |
|  | song\_id INT, |
|  | listen\_time TIMESTAMP |
|  | ); |
|  |  |
|  | INSERT INTO songs\_history VALUES(10011, 777, 1238, 11); |
|  | INSERT INTO songs\_history VALUES(12452, 695, 4520, 1); |
|  |  |
|  | INSERT INTO songs\_weekly VALUES(777, 1238, '2022-08-01 12:00:00'); |
|  | INSERT INTO songs\_weekly VALUES(695, 4520, '2022-08-04 08:00:00'); |
|  | INSERT INTO songs\_weekly VALUES(125, 9630, '2022-08-04 16:00:00'); |
|  | INSERT INTO songs\_weekly VALUES(695, 9852, '2022-08-07 12:00:00'); |
|  |  |
|  | WITH all\_song\_list AS( |
|  | SELECT |
|  | sw.user\_id, |
|  | sw.song\_id, |
|  | count(\*) as song\_plays |
|  | FROM |
|  | songs\_weekly sw |
|  | WHERE |
|  | STR\_TO\_DATE(listen\_time,'%Y-%m-%d') <= STR\_TO\_DATE('08/04/2022','%m/%d/%Y') |
|  | GROUP BY |
|  | sw.user\_id, |
|  | sw.song\_id |
|  |  |
|  | UNION ALL |
|  |  |
|  | SELECT |
|  | sh.user\_id, |
|  | sh.song\_id, |
|  | sh.song\_plays |
|  | FROM |
|  | songs\_history sh |
|  | ) |
|  | SELECT |
|  | user\_id, |
|  | song\_id, |
|  | sum(song\_plays) as song\_plays |
|  | FROM |
|  | all\_song\_list |
|  | GROUP BY |
|  | user\_id, |
|  | song\_id |
|  | ORDER BY |
|  | song\_plays DESC |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE songs\_history; |
|  | DROP TABLE songs\_weekly; |
|  |  |
|  |  |
|  | --Q97: |
|  |  |
|  | CREATE TABLE emails |
|  | ( |
|  | email\_id INT, |
|  | user\_id INT, |
|  | signup\_date TIMESTAMP |
|  | ); |
|  |  |
|  | CREATE TABLE texts |
|  | ( |
|  | text\_id INT, |
|  | email\_id INT, |
|  | signup\_action VARCHAR(20) |
|  | ); |
|  |  |
|  | INSERT INTO emails VALUES(125, 7771, STR\_TO\_DATE('06/14/2022 00:00:00', '%m/%d/%Y %H:%i:%s')); |
|  | INSERT INTO emails VALUES(236, 6950, STR\_TO\_DATE('07/01/2022 00:00:00', '%m/%d/%Y %H:%i:%s')); |
|  | INSERT INTO emails VALUES(433, 1052, STR\_TO\_DATE('07/09/2022 00:00:00', '%m/%d/%Y %H:%i:%s')); |
|  |  |
|  | INSERT INTO texts VALUES(6878, 125, 'Confirmed'); |
|  | INSERT INTO texts VALUES(6920, 236, 'Not Confirmed'); |
|  | INSERT INTO texts VALUES(6994, 236, 'Confirmed'); |
|  |  |
|  | WITH email\_with\_action AS( |
|  | SELECT |
|  | DISTINCT e.email\_id, |
|  | t.signup\_action |
|  | FROM |
|  | emails e |
|  | LEFT JOIN texts t ON e.email\_id = t.email\_id |
|  | ), |
|  | email\_confirm\_table AS( |
|  | SELECT |
|  | email\_id, |
|  | GROUP\_CONCAT(signup\_action ORDER BY signup\_action) AS action, |
|  | position('Confirmed' in GROUP\_CONCAT(signup\_action ORDER BY signup\_action)) as pos |
|  | FROM |
|  | email\_with\_action |
|  | GROUP BY |
|  | email\_id |
|  | ) |
|  | SELECT |
|  | ROUND( |
|  | COUNT( |
|  | CASE |
|  | WHEN pos = 1 |
|  | THEN email\_id |
|  | END |
|  | )\*1.00/ |
|  | count(\*) |
|  | ,2)as confirm\_rate |
|  | FROM email\_confirm\_table |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE emails; |
|  | DROP TABLE texts; |
|  |  |
|  |  |
|  | --Q98: |
|  |  |
|  | CREATE TABLE tweets |
|  | ( |
|  | tweet\_id INT, |
|  | user\_id INT, |
|  | tweet\_date TIMESTAMP |
|  | ); |
|  |  |
|  |  |
|  | INSERT INTO tweets VALUES(214252, 111, STR\_TO\_DATE('06/01/2022 12:00:00', '%m/%d/%Y %H:%i:%s')); |
|  | INSERT INTO tweets VALUES(739252, 111, STR\_TO\_DATE('06/01/2022 12:00:00', '%m/%d/%Y %H:%i:%s')); |
|  | INSERT INTO tweets VALUES(846402, 111, STR\_TO\_DATE('06/02/2022 12:00:00', '%m/%d/%Y %H:%i:%s')); |
|  | INSERT INTO tweets VALUES(241425, 254, STR\_TO\_DATE('06/02/2022 12:00:00', '%m/%d/%Y %H:%i:%s')); |
|  | INSERT INTO tweets VALUES(137374, 111, STR\_TO\_DATE('06/04/2022 12:00:00', '%m/%d/%Y %H:%i:%s')); |
|  |  |
|  |  |
|  | WITH tweet\_per\_day\_by\_user AS |
|  | ( |
|  | SELECT |
|  | user\_id, |
|  | tweet\_date, |
|  | COUNT(\*) as tweet\_count |
|  | FROM |
|  | tweets |
|  | group by |
|  | user\_id, |
|  | tweet\_date |
|  | ) |
|  | SELECT |
|  | user\_id, |
|  | tweet\_date, |
|  | ROUND( |
|  | AVG(tweet\_count) OVER( PARTITION BY user\_id ORDER BY tweet\_date |
|  | ROWS BETWEEN 2 PRECEDING and CURRENT ROW ) |
|  | ,2) as rolling\_avg\_3d |
|  | FROM tweet\_per\_day\_by\_user; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE tweets; |
|  |  |
|  |  |
|  | --Q99: |
|  |  |
|  | CREATE TABLE activities |
|  | ( |
|  | activity\_id INT, |
|  | user\_id INT, |
|  | activity\_type VARCHAR(10), |
|  | time\_spent DECIMAL(5,2), |
|  | activity\_date TIMESTAMP |
|  | ); |
|  |  |
|  | CREATE TABLE age\_breakdown |
|  | ( |
|  | user\_id INT, |
|  | age\_bucket VARCHAR(10) |
|  | ); |
|  |  |
|  | INSERT INTO activities VALUES(7274, 123, 'open', 4.50, STR\_TO\_DATE('06/22/2022 12:00:00', '%m/%d/%Y %H:%i:%s')); |
|  | INSERT INTO activities VALUES(2425, 123, 'send', 3.50, STR\_TO\_DATE('06/22/2022 12:00:00', '%m/%d/%Y %H:%i:%s')); |
|  | INSERT INTO activities VALUES(1413, 456, 'send', 5.67, STR\_TO\_DATE('06/23/2022 12:00:00', '%m/%d/%Y %H:%i:%s')); |
|  | INSERT INTO activities VALUES(1414, 789, 'chat', 11.00, STR\_TO\_DATE('06/25/2022 12:00:00', '%m/%d/%Y %H:%i:%s')); |
|  | INSERT INTO activities VALUES(2536, 456, 'open', 3.00, STR\_TO\_DATE('06/25/2022 12:00:00', '%m/%d/%Y %H:%i:%s')); |
|  |  |
|  | INSERT INTO age\_breakdown VALUES(123, '31-35'); |
|  | INSERT INTO age\_breakdown VALUES(456, '26-30'); |
|  | INSERT INTO age\_breakdown VALUES(789, '21-25'); |
|  |  |
|  | SELECT |
|  | ab.age\_bucket, |
|  | ROUND( |
|  | sum( |
|  | CASE |
|  | WHEN a.activity\_type = 'send' |
|  | THEN |
|  | a.time\_spent |
|  | END |
|  | )\*100.0/sum( |
|  | CASE |
|  | WHEN a.activity\_type in ('open','send') |
|  | THEN |
|  | a.time\_spent |
|  | END |
|  | ) |
|  | ,2) AS send\_perc, |
|  | ROUND( |
|  | sum( |
|  | CASE |
|  | WHEN a.activity\_type = 'open' |
|  | THEN |
|  | a.time\_spent |
|  | END |
|  | )\*100.0/sum( |
|  | CASE |
|  | WHEN a.activity\_type in ('open','send') |
|  | THEN |
|  | a.time\_spent |
|  | END |
|  | ) |
|  | ,2) AS open\_perc |
|  | FROM |
|  | activities a |
|  | JOIN age\_breakdown ab ON a.user\_id = ab.user\_id |
|  | WHERE |
|  | a.activity\_type in ('open','send') |
|  | GROUP BY |
|  | ab.age\_bucket |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE activities; |
|  | DROP TABLE age\_breakdown; |
|  |  |
|  |  |
|  | --Q100: |
|  |  |
|  | CREATE TABLE personal\_profiles |
|  | ( |
|  | profile\_id INT, |
|  | name VARCHAR(30), |
|  | followers INT |
|  | ); |
|  |  |
|  | CREATE TABLE employee\_company |
|  | ( |
|  | personal\_profile\_id INT, |
|  | company\_id INT |
|  | ); |
|  |  |
|  | CREATE TABLE company\_pages |
|  | ( |
|  | company\_id INT, |
|  | name VARCHAR(30), |
|  | followers INT |
|  | ); |
|  |  |
|  | INSERT INTO personal\_profiles VALUES(1, 'Nick Singh', 92000); |
|  | INSERT INTO personal\_profiles VALUES(2, 'Zach Wilson', 199000); |
|  | INSERT INTO personal\_profiles VALUES(3, 'Daliana Liu', 171000); |
|  | INSERT INTO personal\_profiles VALUES(4, 'Ravit Jain', 107000); |
|  | INSERT INTO personal\_profiles VALUES(5, 'Vin Vashishta', 139000); |
|  | INSERT INTO personal\_profiles VALUES(6, 'Susan Wojcicki', 39000); |
|  |  |
|  | INSERT INTO employee\_company VALUES(1, 4); |
|  | INSERT INTO employee\_company VALUES(1, 9); |
|  | INSERT INTO employee\_company VALUES(2, 2); |
|  | INSERT INTO employee\_company VALUES(3, 1); |
|  | INSERT INTO employee\_company VALUES(4, 3); |
|  | INSERT INTO employee\_company VALUES(5, 6); |
|  | INSERT INTO employee\_company VALUES(6, 5); |
|  |  |
|  | INSERT INTO company\_pages VALUES(1 , 'The Data Science Podcast', 8000); |
|  | INSERT INTO company\_pages VALUES(2, 'Airbnb', 700000); |
|  | INSERT INTO company\_pages VALUES(3, 'The Ravit Show', 6000); |
|  | INSERT INTO company\_pages VALUES(4, 'DataLemur', 200); |
|  | INSERT INTO company\_pages VALUES(5, 'YouTube', 16000000); |
|  | INSERT INTO company\_pages VALUES(6, 'DataScience.Vin', 4500); |
|  | INSERT INTO company\_pages VALUES(9, 'Ace The Data Science Interview', 4479); |
|  |  |
|  | WITH profile\_with\_max\_company\_follower AS( |
|  | SELECT |
|  | pp.profile\_id, |
|  | pp.name, |
|  | pp.followers, |
|  | cp.name AS company\_name, |
|  | cp.followers AS company\_follower, |
|  | max(cp.followers) OVER(PARTITION BY pp.profile\_id) as max\_company\_follower |
|  | FROM |
|  | personal\_profiles pp |
|  | JOIN employee\_company ec ON pp.profile\_id = ec.personal\_profile\_id |
|  | JOIN company\_pages cp ON cp.company\_id = ec.company\_id |
|  | ORDER BY |
|  | pp.name |
|  | ) |
|  | SELECT |
|  | DISTINCT profile\_id |
|  | FROM |
|  | profile\_with\_max\_company\_follower |
|  | WHERE |
|  | followers > max\_company\_follower |
|  | ORDER BY |
|  | profile\_id |
|  | ; |
|  |  |
|  | --drop tables |
|  |  |
|  | DROP TABLE personal\_profiles; |
|  | DROP TABLE employee\_company; |
|  | DROP TABLE company\_pages; |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |