

SQL

Using the three data sets we've provided to you, please provide us with queries, in MySQL syntax, that answer the following questions:

1. Using the event_table, how many events were published in September 2014?

Total, 182 events were published in September 2014.

Query:

```
SELECT count(distinct(event_id))
FROM events
WHERE event_publish_date >= "2014-09-01"
      AND event_publish_date <= "2014-09-30"
;
```

2. What are the Top 10 organizer ids by the number of free events created outside the USA?

Query:

```
SELECT
    o.organizer_id,
    count(distinct(event_id)) as event_count
FROM organizers o
INNER JOIN events e ON o.organizer_id = e.organizer_id
WHERE signup_country != "USA"
      AND event_paid_type = "free event"
GROUP BY organizer_id, signup_country, event_paid_type
ORDER BY event_count DESC
LIMIT 10
;
```

3. How many organizers had over 2 events that started within the last 15 months?

I started out using this query to look at a list of the organizer IDs with over 2 events starting within the past 15 months, which gave me 88 results:

```
SELECT
    organizer_id,
    count(distinct(event_id)) as event_count
FROM events
WHERE event_start_date >= DATE_ADD(current_timestamp(), INTERVAL -15 MONTH)
GROUP BY organizer_id
HAVING event_count > 2
ORDER BY event_count DESC
;
```

If you wanted to, with a nested query you can get the answer immediately, instead of as a table.

4. How many organizer who signed up in the USA had at least one order?

1,736 organizers who signed up in the United States had at least one order.

Query:

```
SELECT
COUNT(DISTINCT(organizer))
  FROM(
    SELECT
      e.organizer_id AS organizer,
      COUNT(DISTINCT(order_id)) AS order_count
    FROM organizers o
    INNER JOIN events e on o.organizer_id = e.organizer_id
    INNER JOIN orders od on od.event_id = e.event_id
    WHERE o.signup_country = "USA"
    GROUP BY o.organizer_id
    HAVING order_count >= 1
  ) AS a
;
```

5. How many organizers had a least one paid order but never a free order?

697 organizers had at least one paid order but never a free order, as shown by this query:

```
SELECT
COUNT(DISTINCT(CASE WHEN paid_event_count > 0.0 AND free_event_count = 0.0 THEN
organizer ELSE NULL END))
  FROM(
    SELECT
      o.organizer_id AS organizer,
      COUNT(DISTINCT(CASE WHEN order_revenue = 0.0 THEN e.event_id ELSE null END))
AS free_event_count,
      COUNT(DISTINCT(CASE WHEN order_revenue > 0.0 THEN e.event_id ELSE null END))
AS paid_event_count
    FROM
      organizers as o
      inner join events as e on e.organizer_id = o.organizer_id
      inner join orders as od on e.event_id = od.event_id
    GROUP BY organizer
  ) AS a
;
```

6. What was the total order revenue for Music category events that are public, and started in 2016?

The query below lists all event categories and their total order revenue for all public events created

since January 1, 2016. From this table, public Music category events have made \$636.00 since 1/1/2016.

```
SELECT
    event_category,
    ROUND(SUM(order_revenue),2) AS total_order_revenue
FROM events AS e
INNER JOIN orders AS o ON e.event_id = o.event_id
WHERE is_listed = "public"
AND event_start_date >= "2016-01-01"
GROUP BY event_category
;
```

If you wanted a query that would *only* display events from the music event category, just put that in the WHERE clause like so:

```
SELECT
    event_category,
    ROUND(SUM(order_revenue),2) AS total_order_revenue
FROM events AS e
INNER JOIN orders AS o ON e.event_id = o.event_id
WHERE is_listed = "public"
AND event_category = "Music"
AND event_start_date >= "2016-01-01"
GROUP BY event_category
;
```

1. Using in the all three tables, please design a query which the output captures all the organizer_ids organizer_table. The query should also include the following fields:

- a. Organizer_id
- b. signup country broken out by USA vs. Rest of the world
- c. organizer type broken out by Paid vs. Free
- d. # of events the organizer published since 2014
- e. # of lifetime orders
- f. # of lifetime tickets
- g. # of lifetime free tickets sold
- h. # of lifetime public events published

This query displays in a table all organizer_ids. It contains about a hundred more rows than the organizers table, because some organizers may have put on both free and paid events, or put on

events both in the USA and in the rest of the world.

```
SELECT
    o.organizer_id,
    CASE WHEN signup_country = "USA" THEN "USA" ELSE "Rest of the World" END AS
signup_region,
    event_paid_type,
    COUNT(DISTINCT(CASE WHEN event_publish_date >= "2014-01-01" THEN e.event_id
ELSE NULL END )) AS events_since_2014,
    COUNT(DISTINCT(od.order_id)) AS lifetime_orders,
    SUM(total_tickets) AS lifetime_tickets,
    SUM(CASE WHEN order_revenue = 0 THEN total_tickets ELSE 0 END) AS
lifetime_free_tickets,
    COUNT(DISTINCT(CASE WHEN is_listed = "public" THEN e.event_id ELSE NULL END))
AS lifetime_public_events_published
FROM organizers AS o
LEFT JOIN events AS e ON o.organizer_id = e.organizer_id
LEFT JOIN orders AS od ON e.event_id = od.event_id
GROUP BY o.organizer_id, signup_region, event_paid_type
ORDER BY o.organizer_id, signup_region, event_paid_type
;
```

Analysis

Looking at the dataset we provided, please pick out a few interesting observations or trends. We are most interested in seeing your thought process and creativity in approaching an openended data problem. Feel free to include plots and tables as needed.

Please refer to attached Analysis file.