Marathon Project: Milestone 3 Business Report

Summary:

This business report summarizes insights regarding race attendance and financials from 2023, specifically regarding which events had the highest expenses, which events had insurance expenses greater than average, and how many small sponsorships individual events received.

Topic #1: 5 Most Expensive Events

The 5 most expensive events in descending order were the Chi-town Dash Race, Philadelphia Race, Surfs Up Race, Indy Race, and the Silicon Valley Sprint Race. They occurred in 2022 in the cities of Chicago, Philadelphia, San Diego, Indianapolis, and San Jose with total expenses of $1,382,841, $1,034,325, $972,184, $669,760, and $637,673 respectively.

SQL code:

```sql
// Main Query Part B1

SELECT
    re.event_name AS event_name,
    re.event_date AS event_date,
    re.city AS city,
    re.state AS state,
    SUM(ee.expense_amount) AS total_expenses
FROM
    RACE_EVENTS re
JOIN
    EVENT_EXPENSES ee
ON
    re.event_id = ee.event_id
GROUP BY
    re.event_name, re.event_date, re.city, re.state
ORDER BY
    total_expenses DESC
LIMIT 5;
```

Results:

Results | Chart

	EVENT_NAME	EVENT_DATE	CITY	STATE	TOTAL_EXPENSES
1	Chi-town Dash Race Weekend	2022-08-21	Chicago	IL	1382841.69
2	Philadelphia Race Weekend	2022-06-19	Philadelphia	PA	1034325.88
3	Surfs Up Race Weekend	2022-06-26	San Diego	CA	972184.02
4	Indy Race Weekend	2022-07-10	Indianapolis	IN	669760.99
5	Silicon Valley Sprint Race Weekend	2022-11-06	San Jose	CA	637673.3

These results suggest certain events which cater more heavily to a casual audience may be more expensive than typical events. We have a dash race, a sprint race, an Indy race, and a Surfs Up race, all of which sound like they could be more about fun than competition. This could lead to far more expensive venues.

Topic #2: Abnormal Insurance Expenses

This table provides the event name, date on which the event occurred, and the insurance expense associated with it for events which had insurance expenses above the average.

SQL code:

```sql
// Main Query Part B2

SELECT
    re.event_name AS event_name,
    re.event_date AS event_date,
    ee.expense_amount AS insurance_expense
FROM
    RACE_EVENTS re
JOIN
    EVENT_EXPENSES ee
ON
    re.event_id = ee.event_id
WHERE
    ee.expense_type = 'expense_insurance'
    AND ee.expense_amount > (
        SELECT AVG(expense_amount)
        FROM EVENT_EXPENSES
        WHERE expense_type = 'expense_insurance'
    )
ORDER BY
    insurance_expense DESC
LIMIT 10;
```

Results:

Results | Chart

	EVENT_NAME	EVENT_DATE	INSURANCE_EXPENSE
1	Chi-town Dash Race Weekend	2022-08-21	464675.93
2	Surfs Up Race Weekend	2022-06-26	314351.34
3	Philadelphia Race Weekend	2022-06-19	286139.88
4	Sprint the Sound Race Weekend	2022-06-05	161465.45
5	Silicon Valley Sprint Race Weekend	2022-11-06	158692.29
6	Music City Race Weekend	2022-06-12	150216.74
7	Boston Tea Party Race Weekend	2022-07-17	135794.03
8	Indy Race Weekend	2022-07-10	127093.62
9	Miami International Race Weekend	2022-02-27	123552.92
10	Research Triangle Race Weekend	2022-12-04	119374.52

These results suggest insurance expenses may play a significant role in the total expense for an event. There are a total of 8 expense types assigned to each event, and in the Chi-town Dash Race for example, insurance expenses account for approximately a third of total expenses. Time should be taken to look into why insurance costs were so high on these events and if that could be due to something around fun events.

Topic #3: Event Small Sponsorship Counts

This query provides us with the event id, the count of small sponsorships for that event, and the value of those sponsorships. It would be useful for looking at small business support of events in different cities as well as other things.

SQL code:

```sql
// Main Query Part B3

SELECT
    es.event_id AS event_id,
    COUNT(*) AS small_sponsorship_count,
    SUM(es.sponsor_amount) AS total_sponsorship_value
FROM
    EVENT_SPONSORS es
WHERE
    (es.sponsor_level = 'Bronze' AND es.employee_count = '1-9')
    OR
    (es.sponsor_level IN ('Bronze', 'Silver') AND es.employee_count = '10-99')
GROUP BY
    es.event_id
ORDER BY
    small_sponsorship_count DESC
LIMIT 10;
```

Results:

Results | Chart

	EVENT_ID	SMALL_SPONSORSHIP_COUNT	TOTAL_SPONSORSHIP_VALUE
1	87418	73	66000
2	67203	45	40500
3	19532	42	31500
4	73072	36	31500
5	52843	25	21000
6	22047	20	17250
7	55917	18	20250
8	15208	18	22500
9	90893	18	15750
10	44171	16	16500

These results suggest whichever events are represented by event id's 87418, 67203, 19532, and 73072 had very high turnouts of small sponsors. This likely means that these events either occurred in large cities, had some factor driving small businesses towards them, or that these cities simply have especially strong business communities. These would be useful to investigate to try to drive more support from and for small businesses in the future.

Text SQL Code:

```
// Main Query Part B1

SELECT
  re.event_name AS event_name,
  re.event_date AS event_date,
  re.city AS city,
  re.state AS state,
  SUM(ee.expense_amount) AS total_expenses
FROM
  RACE_EVENTS re
JOIN
  EVENT_EXPENSES ee
ON
  re.event_id = ee.event_id
GROUP BY
  re.event_name, re.event_date, re.city, re.state
ORDER BY
  total_expenses DESC
LIMIT 5;
```

```
// Main Query Part B2


SELECT
  re.event_name AS event_name,
  re.event_date AS event_date,
  ee.expense_amount AS insurance_expense
FROM
  RACE_EVENTS re
JOIN
  EVENT_EXPENSES ee
ON
  re.event_id = ee.event_id
WHERE
  ee.expense_type = 'expense_insurance'
  AND ee.expense_amount > (
    SELECT AVG(expense_amount)
    FROM EVENT_EXPENSES
    WHERE expense_type = 'expense_insurance'
  )
ORDER BY
  insurance_expense DESC
LIMIT 10;
```

```
// Main Query Part B3


SELECT
  es.event_id AS event_id,
  COUNT(*) AS small_sponsorship_count,
  SUM(es.sponsor_amount) AS total_sponsorship_value
FROM
  EVENT_SPONSORS es
WHERE
  (es.sponsor_level = 'Bronze' AND es.employee_count = '1-9')
  OR
  (es.sponsor_level IN ('Bronze', 'Silver') AND es.employee_count = '10-99')
GROUP BY
  es.event_id
ORDER BY
  small_sponsorship_count DESC
LIMIT 10;
```