PT1 Stage 3 Regrade (Changes)

1. We added the screenshots for the 1000+ rows for 3 tables.

-- Event, ***>1000 rows from Ticketmaster API***

```
CREATE TABLE IF NOT EXISTS Event (
  event_id BIGINT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(255) NOT NULL,
  date DATE,
  ticket_price DECIMAL(10,2),
  venue_name VARCHAR(255) NOT NULL,
  venue_id BIGINT
  FOREIGN KEY (venue_id) REFERENCES Venue(venue_id)
  );
```

```
mysql> SELECT COUNT(*) AS row_count
    -> FROM Event;
+-----------+
| row_count |
+-----------+
|      1188 |
+-----------+
1 row in set (0.00 sec)
```

-- AirbnbListing, ***>1000 rows from InsideAirbnb CSV***

```
CREATE TABLE IF NOT EXISTS AirbnbListing (
  listing_id BIGINT PRIMARY KEY,
  city_id INT NOT NULL,
  latitude DECIMAL(9,6),
  longitude DECIMAL(9,6),
  price_per_night DECIMAL(10,2),
  availability_365 INT,
  room_type VARCHAR(50),
  city_name VARCHAR(255) NOT NULL
  FOREIGN KEY (city_id) REFERENCES City(city_id)
);
```

```
mysql> SELECT COUNT(*) AS row_count
    -> FROM AirbnbListing;
+-----------+
| row_count |
+-----------+
|      7071 |
+-----------+
1 row in set (0.01 sec)
```

-- Nearby (Event <-> Airbnb), ***> 1000 rows, Populated from Event and Airbnb***

```
CREATE TABLE IF NOT EXISTS Nearby (
  event_id BIGINT,
  listing_id BIGINT,
  total_cost DECIMAL(12,2),
  distance DECIMAL(8,2),
  PRIMARY KEY (event_id, listing_id),
  FOREIGN KEY (event_id) REFERENCES Event(event_id),
  FOREIGN KEY (listing_id) REFERENCES AirbnbListing(listing_id)
);
```

```
mysql> SELECT COUNT(*) AS row_count
    -> FROM Nearby ;
+-----------+
| row_count |
+-----------+
|   8400348 |
+-----------+
1 row in set (9.44 sec)
```

2. We edited Query 3 to include JOIN and Aggregation via GROUP BY.

QUERY 3: Events with most available Airbnb Listings (within 5 miles)

- Focuses on listing availability count
- Concepts: JOIN, Aggregation via Group by

```
SELECT E.event_id, E.name AS event_name, C.city_name, C.state, COUNT(A.listing_id) AS
num_available_listings, ROUND(AVG(A.price_per_night), 2) AS avg_price_per_night,
MIN(N.distance) AS closest_listing_distance
FROM Event E
JOIN Venue V ON E.venue_id = V.venue_id
JOIN City C ON V.city_id = C.city_id
JOIN Nearby N ON E.event_id = N.event_id
JOIN AirbnbListing A ON N.listing_id = A.listing_id
WHERE N.distance <= 5 AND A.availability_365 > 0
GROUP BY E.event_id, E.name, C.city_name, C.state
ORDER BY num_available_listings DESC, avg_price_per_night ASC
LIMIT 15;
```

event_id	event_name	city_name	state	num_available_listings	avg_price_per_night	closest_listing_distance
60	Chicago Blackhawks vs. Ottawa Senators	Chicago	Illinois	4771	534.35	0.15
213	Chicago Blackhawks vs. New Jersey Devils	Chicago	Illinois	4771	534.35	0.15
270	Chicago Blackhawks vs. Calgary Flames	Chicago	Illinois	4771	534.35	0.15
287	Chicago Blackhawks vs. Seattle Kraken	Chicago	Illinois	4771	534.35	0.15
740	Chicago Blackhawks vs. St. Louis Blues	Chicago	Illinois	4771	534.35	0.15
330	Chicago Blackhawks vs. Colorado Avalanche	Chicago	Illinois	4771	534.35	0.15
336	PAUL McCARTNEY: GOT BACK	Chicago	Illinois	4771	534.35	0.15
345	PAUL McCARTNEY: GOT BACK	Chicago	Illinois	4771	534.35	0.15
357	Chicago Blackhawks vs. Minnesota Wild	Chicago	Illinois	4771	534.35	0.15
374	Chicago Blackhawks vs. Nashville Predators	Chicago	Illinois	4771	534.35	0.15
399	Chicago Blackhawks vs. Anaheim Ducks	Chicago	Illinois	4771	534.35	0.15
513	Chicago Blackhawks vs. Detroit Red Wings	Chicago	Illinois	4771	534.35	0.15
721	Chicago Blackhawks vs. Vegas Golden Knights	Chicago	Illinois	4771	534.35	0.15
666	Chicago Blackhawks vs. Pittsburgh Penguins	Chicago	Illinois	4771	534.35	0.15
627	Chicago Blackhawks vs. Philadelphia Flyers	Chicago	Illinois	4771	534.35	0.15

15 rows in set (4.35 sec)

Indexing

Index	Attributes Indexed	Query Cost	Notes
Baseline	N/A	83,755	N/A
idx_nearby	Nearby (distance, event_id)	44,283	Improvement
idx_airbnb	AirbnbListing (availability_365)	83,755	No improvement
idx_event	Event (venue_id)	83,755	No improvement
idx_venue	Venue (city_id)	83,755	No improvement

Indexing Nearby(distance, event_id) reduced the query cost from 83,755 to 44,283, which was the most significant performance improvement for this query. The index helps to efficiently filter listings within 5 miles and join on N.event_id, so we avoid a full scan of the Nearby table. Without it, the database must check all rows, which is costly given the large number of event-listing pairs. The other indexes did not improve performance, as the filters and joins they support are either not highly selective or already handled. Overall, the index on Nearby directly supports the query's main filtering and join conditions, resulting in the largest cost reduction.