

# Sub Queries and CTE in MySQL

## SQL Queries for Analyzing Athlete Data and Patient Claims: Olympic Performance and Insurance Insights

### Olympics Dataset Queries

#### 1. Athletes Winning Gold in 2008 with Above-Average Height

```
SELECT Name
FROM olympics
WHERE Medal = 'Gold'
AND Height > (SELECT AVG(Height) FROM olympics WHERE Year = 2008)
AND Year = 2008;
```

**Description:** Finds athletes who won a gold medal in the 2008 Olympics and are taller than the average height.

---

#### 2. Basketball Medalists in 2016 with Below-Average Weight

```
SELECT Name, Medal
FROM olympics
WHERE Weight < (SELECT AVG(Weight)
                FROM olympics
                WHERE Year = 2016 AND Medal IS NOT NULL)
AND Year = 2016
AND Medal IS NOT NULL
AND Sport = 'Basketball';
```

**Description:** Lists athletes who won a basketball medal in 2016 and weigh less than the average medalist.

---

#### 3. Swimmers Winning Medals in Both 2008 and 2016

```
SELECT Name, Medal
FROM olympics
WHERE Sport = 'Swimming'
AND Medal IS NOT NULL
AND Name IN (SELECT Name FROM olympics WHERE Year = 2016 AND Sport =
'Swimming' AND Medal IS NOT NULL)
```

@BhoopendraVishwakarma

```
AND Name IN (SELECT Name FROM olympics WHERE Year = 2008 AND Sport =
'Swimming' AND Medal IS NOT NULL);
```

**Description:** Displays swimmers who won medals in both the 2008 and 2016 Olympics.

---

#### 4. Countries Winning More Than 50 Medals in a Single Year

```
WITH MedalCounts AS (
    SELECT Country, Year, COUNT(Medal) AS MedalCount
    FROM olympics
    WHERE Medal IS NOT NULL
    GROUP BY Country, Year
)
SELECT DISTINCT Country
FROM MedalCounts
WHERE MedalCount > 50;
```

**Description:** Lists countries with over 50 medals in any year.

---

#### 5. Athletes Winning Medals in Multiple Sports in the Same Year

```
WITH MoreSports AS (
    SELECT Name
    FROM olympics
    WHERE Medal IS NOT NULL
    GROUP BY Year, Name
    HAVING COUNT(DISTINCT Sport) > 1
)
SELECT Name
FROM MoreSports;
```

**Description:** Retrieves athletes who won medals in more than one sport in the same year.

---

#### 6. Average Weight Difference Between Male and Female Medalists

```
WITH GenderWeights AS (
    SELECT Event, Sex, AVG(Weight) AS AvgWeight
    FROM olympics
    WHERE Medal IS NOT NULL
    GROUP BY Event, Sex
),
EventWeightDiff AS (
    SELECT Event, ABS(MAX(CASE WHEN Sex = 'M' THEN AvgWeight ELSE NULL END) -
MAX(CASE WHEN Sex = 'F' THEN AvgWeight ELSE NULL END)) AS WeightDifference
    FROM GenderWeights
    GROUP BY Event
```

```

HAVING COUNT(DISTINCT Sex) = 2
)
SELECT AVG(WeightDifference) AS AverageWeightDifference
FROM EventWeightDiff;

```

**Description:** Calculates the average weight difference between male and female medalists in the same event.

---



## Insurance Dataset Queries

### 1. Patients Claiming Above-Average Amount (Smokers with Children in Southeast)

```

SELECT COUNT(*) AS Patients
FROM insurance
WHERE Claim > (SELECT AVG(Claim)
                FROM insurance
                WHERE Smoker = 'Yes' AND Children >= 1 AND Region =
'Southeast' AND Claim IS NOT NULL);

```

**Description:** Counts patients who claim more than the average smoker with children in the Southeast.

---

### 2. Patients with BMI Above Average and High Claims (Non-Smokers)

```

SELECT COUNT(*)
FROM insurance
WHERE BMI > (SELECT AVG(BMI) FROM insurance WHERE Children >= 1)
AND Claim > (SELECT AVG(Claim) FROM insurance WHERE Smoker = 'No');

```

**Description:** Finds non-smokers with higher-than-average BMI and claims.

---

### 3. Patients with BMI Above Average for Diabetics in Southwest

```

SELECT COUNT(*)
FROM insurance
WHERE Claim > (SELECT AVG(Claim)
                FROM insurance
                WHERE BMI > (SELECT AVG(BMI)
                              FROM insurance
                              WHERE Diabetic = 'Yes' AND Children >= 1 AND
Region = 'Southwest')
                AND Claim IS NOT NULL);

```

**Description:** Counts patients with high BMI and claims in the diabetic group in Southwest.

---

#### 4. Average Claim Difference Between Smokers and Non-Smokers

##### Query 1 (Using Subqueries):

```
WITH Difference AS (  
    SELECT ABS(  
        (SELECT AVG(Claim)  
         FROM insurance  
         WHERE Smoker = 'Yes' AND BMI = i.BMI AND Children = i.Children AND  
         Claim IS NOT NULL)  
        -  
        (SELECT AVG(Claim)  
         FROM insurance  
         WHERE Smoker = 'No' AND BMI = i.BMI AND Children = i.Children AND  
         Claim IS NOT NULL)  
    ) AS ClaimDifference  
    FROM insurance i  
)  
SELECT ClaimDifference  
FROM Difference  
WHERE ClaimDifference IS NOT NULL;
```

##### Query 2 (Using Grouped Aggregation):

```
WITH GroupedClaims AS (  
    SELECT BMI, Children,  
        AVG(CASE WHEN Smoker = 'Yes' THEN Claim END) AS AvgClaimSmokers,  
        AVG(CASE WHEN Smoker = 'No' THEN Claim END) AS AvgClaimNonSmokers  
    FROM insurance  
    WHERE Claim IS NOT NULL  
    GROUP BY BMI, Children  
)  
,  
Difference AS (  
    SELECT ABS(AvgClaimSmokers - AvgClaimNonSmokers) AS ClaimDifference  
    FROM GroupedClaims  
    WHERE AvgClaimSmokers IS NOT NULL AND AvgClaimNonSmokers IS NOT NULL  
)  
SELECT ClaimDifference  
FROM Difference;
```

**Description:** Calculates the claim difference between smokers and non-smokers for patients with the same BMI and number of children.

---

#### 🌟 Key Highlights

- Efficient use of **CTEs** (`WITH` clause) to simplify complex queries.
- **Subqueries** and **aggregations** to calculate averages and differences.
- Logical use of `HAVING` for filtering grouped data.
- Queries are structured for **clear problem-solving** and **better performance**.

Thanks for looking with **patience**

@BhoopendraVishwakarma