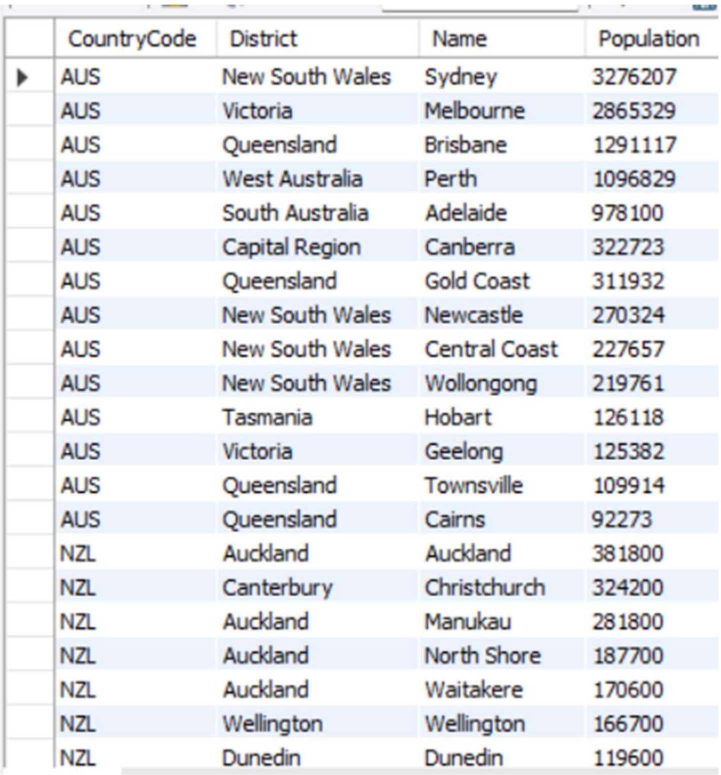
Totals and Subtotals

```
-- Create a new database
CREATE DATABASE IF NOT EXISTS city_population_Subtotals;
USE city_population_Subtotals;

-- Create a table for City
CREATE TABLE IF NOT EXISTS City (
    CountryCode CHAR(3),
    District VARCHAR(255),
    Name VARCHAR(255),
    Population INT
);

-- Insert values into the City table
INSERT INTO City (CountryCode, District, Name, Population) VALUES
    ('AUS', 'New South Wales', 'Sydney', 3276207),
    ('AUS', 'Victoria', 'Melbourne', 2865329),
    ('AUS', 'Queensland', 'Brisbane', 1291117),
    ('AUS', 'West Australia', 'Perth', 1096829),
    ('AUS', 'South Australia', 'Adelaide', 978100),
    ('AUS', 'Capital Region', 'Canberra', 322723),
    ('AUS', 'Queensland', 'Gold Coast', 311932),
    ('AUS', 'New South Wales', 'Newcastle', 270324),
    ('AUS', 'New South Wales', 'Central Coast', 227657),
    ('AUS', 'New South Wales', 'Wollongong', 219761),
    ('AUS', 'Tasmania', 'Hobart', 126118),
    ('AUS', 'Victoria', 'Geelong', 125382),
('AUS', 'Queensland', 'Townsville', 109914),
('AUS', 'Queensland', 'Cairns', 92273),
('NZL', 'Auckland', 'Auckland', 381800),
('NZL', 'Canterbury', 'Christchurch', 324200),
('NZL', 'Auckland', 'Manukau', 281800),
('NZL', 'Auckland', 'North Shore', 187700),
('NZL', 'Auckland', 'Waitakere', 170600),
('NZL', 'Wellington', 'Wellington', 166700),
('NZL', 'Dunedin', 'Dunedin', 119600),
('NZL', 'Hamilton', 'Hamilton', 117100),
('NZL', 'Wellington', 'Lower Hutt', 98100);
```

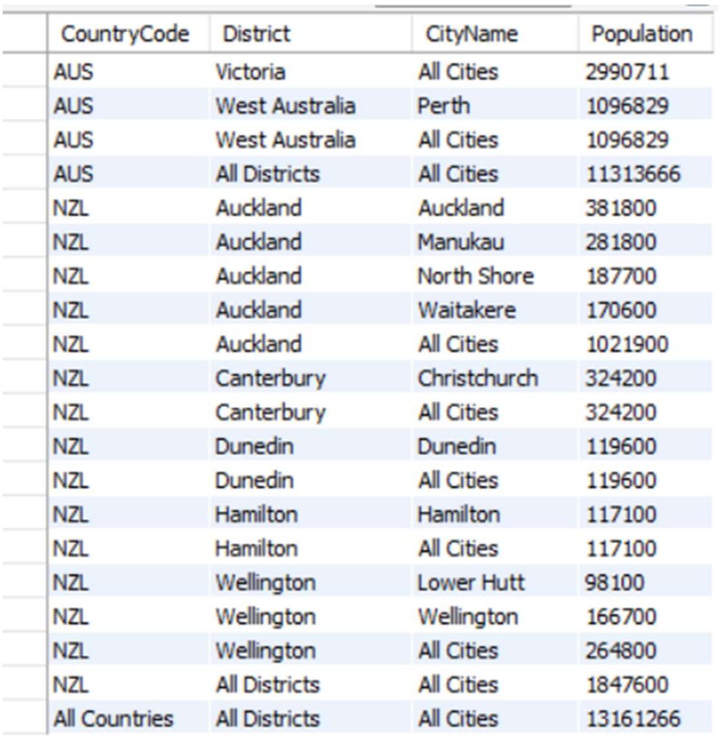
```
-- Query to select all records from the City table

SELECT

   CountryCode,

   District,

   Name,

   Population

FROM City

WHERE CountryCode IN ('AUS', 'NZL');
```



CountryCode	District	Name	Population
AUS	New South Wales	Sydney	3276207
AUS	Victoria	Melbourne	2865329
AUS	Queensland	Brisbane	1291117
AUS	West Australia	Perth	1096829
AUS	South Australia	Adelaide	978100
AUS	Capital Region	Canberra	322723
AUS	Queensland	Gold Coast	311932
AUS	New South Wales	Newcastle	270324
AUS	New South Wales	Central Coast	227657
AUS	New South Wales	Wollongong	219761
AUS	Tasmania	Hobart	126118
AUS	Victoria	Geelong	125382
AUS	Queensland	Townsville	109914
AUS	Queensland	Cairns	92273
NZL	Auckland	Auckland	381800
NZL	Canterbury	Christchurch	324200
NZL	Auckland	Manukau	281800
NZL	Auckland	North Shore	187700
NZL	Auckland	Waitakere	170600
NZL	Wellington	Wellington	166700
NZL	Dunedin	Dunedin	119600

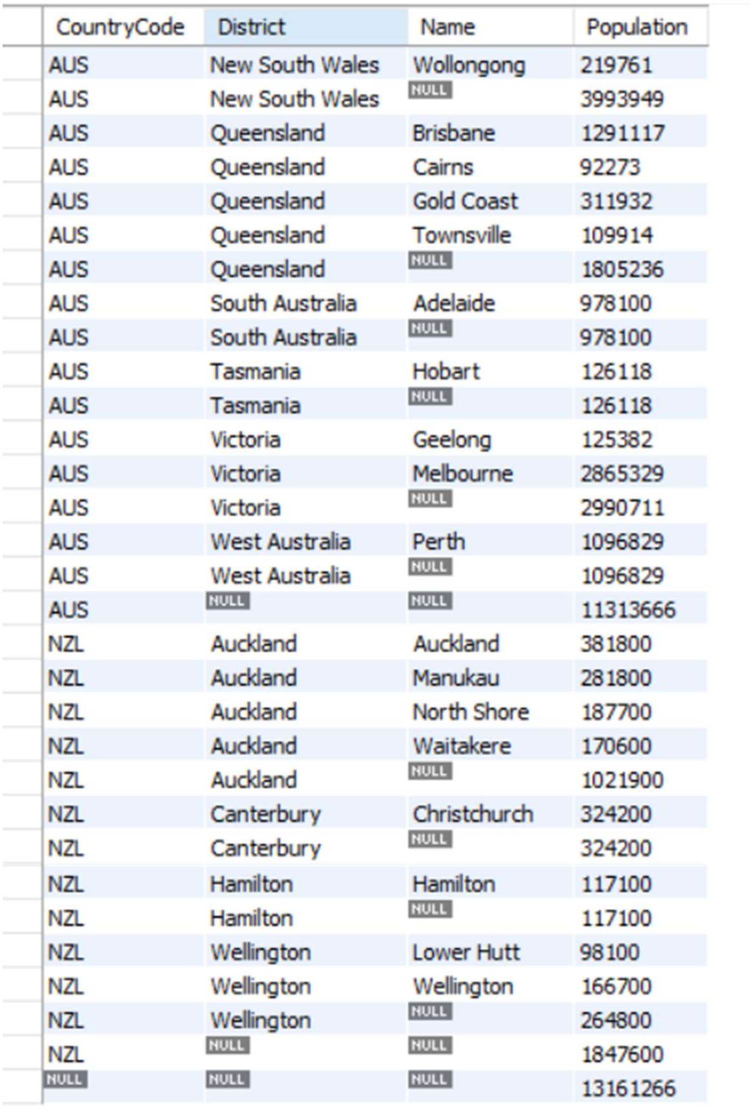
```
-- Query to retrieve city populations with subtotals and grand total

SELECT

   IF(GROUPING(CountryCode), 'All Countries', CountryCode) AS CountryCode,

   IF(GROUPING(District), 'All Districts', District) AS District,wht

   IF(GROUPING(Name), 'All Cities', Name) As CityName,

   SUM(Population) AS Population

FROM City

WHERE CountryCode IN ('AUS', 'NZL')
```

```
GROUP BY CountryCode, District, Name WITH ROLLUP;
```



CountryCode	District	CityName	Population
AUS	Victoria	All Cities	2990711
AUS	West Australia	Perth	1096829
AUS	West Australia	All Cities	1096829
AUS	All Districts	All Cities	11313666
NZL	Auckland	Auckland	381800
NZL	Auckland	Manukau	281800
NZL	Auckland	North Shore	187700
NZL	Auckland	Waitakere	170600
NZL	Auckland	All Cities	1021900
NZL	Canterbury	Christchurch	324200
NZL	Canterbury	All Cities	324200
NZL	Dunedin	Dunedin	119600
NZL	Dunedin	All Cities	119600
NZL	Hamilton	Hamilton	117100
NZL	Hamilton	All Cities	117100
NZL	Wellington	Lower Hutt	98100
NZL	Wellington	Wellington	166700
NZL	Wellington	All Cities	264800
NZL	All Districts	All Cities	1847600
All Countries	All Districts	All Cities	13161266

```
-- Query without labels (using NULLs) to get the same output

SELECT

   CountryCode,

   District,

   Name,

   SUM(Population) AS Population

FROM City

WHERE CountryCode IN ('AUS', 'NZL')

GROUP BY CountryCode, District, Name WITH ROLLUP;
```



CountryCode	District	Name	Population
AUS	New South Wales	Wollongong	219761
AUS	New South Wales	NULL	3993949
AUS	Queensland	Brisbane	1291117
AUS	Queensland	Cairns	92273
AUS	Queensland	Gold Coast	311932
AUS	Queensland	Townsville	109914
AUS	Queensland	NULL	1805236
AUS	South Australia	Adelaide	978100
AUS	South Australia	NULL	978100
AUS	Tasmania	Hobart	126118
AUS	Tasmania	NULL	126118
AUS	Victoria	Geelong	125382
AUS	Victoria	Melbourne	2865329
AUS	Victoria	NULL	2990711
AUS	West Australia	Perth	1096829
AUS	West Australia	NULL	1096829
AUS	NULL	NULL	11313666
NZL	Auckland	Auckland	381800
NZL	Auckland	Manukau	281800
NZL	Auckland	North Shore	187700
NZL	Auckland	Waitakere	170600
NZL	Auckland	NULL	1021900
NZL	Canterbury	Christchurch	324200
NZL	Canterbury	NULL	324200
NZL	Hamilton	Hamilton	117100
NZL	Hamilton	NULL	117100
NZL	Wellington	Lower Hutt	98100
NZL	Wellington	Wellington	166700
NZL	Wellington	NULL	264800
NZL	NULL	NULL	1847600
NULL	NULL	NULL	13161266

snowflaking (snowflake schema)

Snowflaking, or the snowflake schema, is a data modeling technique used in data warehousing. It involves normalizing dimension tables to reduce redundancy and improve data integrity. In a snowflake schema, dimensions are stored in multiple related tables, creating a pattern that resembles a snowflake.

Key points about the snowflake schema:

Normalization: The snowflake schema normalizes dimension tables. Normalization involves breaking down tables into smaller, related tables to eliminate redundancy. This process makes data maintenance easier and reduces the risk of data integrity issues.

Star Schema vs. Snowflake Schema:

Star Schema: In a star schema, there is a central fact table connected to denormalized dimension tables. It's designed for simplicity and query performance.

Snowflake Schema: In a snowflake schema, dimensions are further normalized into multiple related tables. This reduces redundancy but can result in more complex queries due to additional joins.

Example:

Star Schema (simplified): Fact table connected to denormalized dimension tables.

```
+-----------+   +-------------+   +----------+
| Fact      |   | Dim1        |   | Dim2     |
| Table     |   | Table       |   | Table    |
+-----------+   +-------------+   +----------+
```

Snowflake Schema (simplified): Fact table connected to normalized dimension tables.

```
+-----------+   +-------------+   +------------+
| Fact      |   | Dim1        |   | Dim2       |
| Table     |   | Table       |   | Table      |
+-----------+   +-------------+   +------------+
                |
          +--------------+
          | Dim1 Detail  |
          | Table        |
          +--------------+
```

Purpose of Snowflake Schema:

Reduce Redundancy: Snowflaking reduces redundant data, making the schema less prone to data integrity issues.

Simplify Data Maintenance: Easier management and maintenance of data due to normalized structures.

Save Disk Space: Requires less disk space compared to highly denormalized structures.

Considerations:

Query Performance: Snowflake schema may introduce additional joins, impacting query performance compared to star schema. However, the impact might be acceptable based on specific requirements and workload.

Starflake Schema: Sometimes, a schema is referred to as a "starflake schema" when it combines characteristics of both star and snowflake schemas. This occurs when at least one dimension is normalized.

Use Cases for Snowflake Schema:

Sparse Attributes: When dimensions have sparsely populated attributes with mostly NULL values.

Many-to-Many Relationships: To support many-to-many relationships and limit instances.

Large Dimensions with Redundancy: In large dimensions with redundant data, especially in low cardinality attributes.

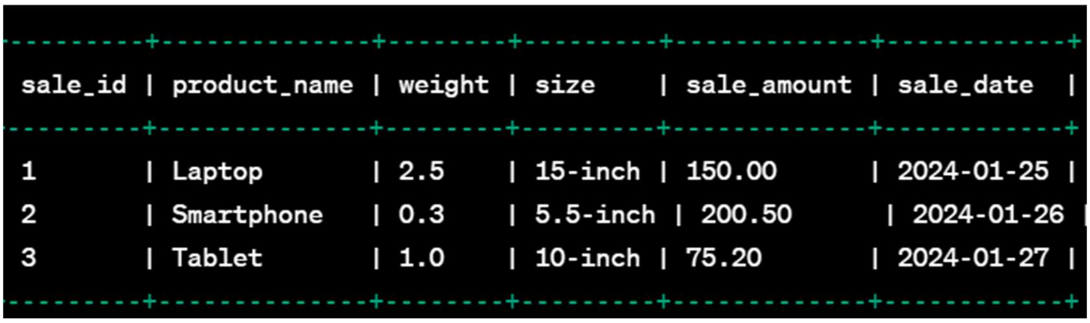
In summary, while snowflaking may introduce complexity in queries, it offers benefits in terms of reduced redundancy, improved data integrity, and efficient data maintenance in data warehousing scenarios. The decision to use a snowflake schema depends on specific use cases, query patterns, and the characteristics of the data being stored.

```
-- Create Sales Table

CREATE TABLE sales (

    sale_id INT PRIMARY KEY,

    product_id INT,

    sale_amount DECIMAL(10, 2),

    sale_date DATE

);


-- Insert sample data into sales table

INSERT INTO sales (sale_id, product_id, sale_amount, sale_date)

VALUES

    (1, 101, 150.00, '2024-01-25'),

    (2, 102, 200.50, '2024-01-26'),

    (3, 103, 75.20, '2024-01-27');
```

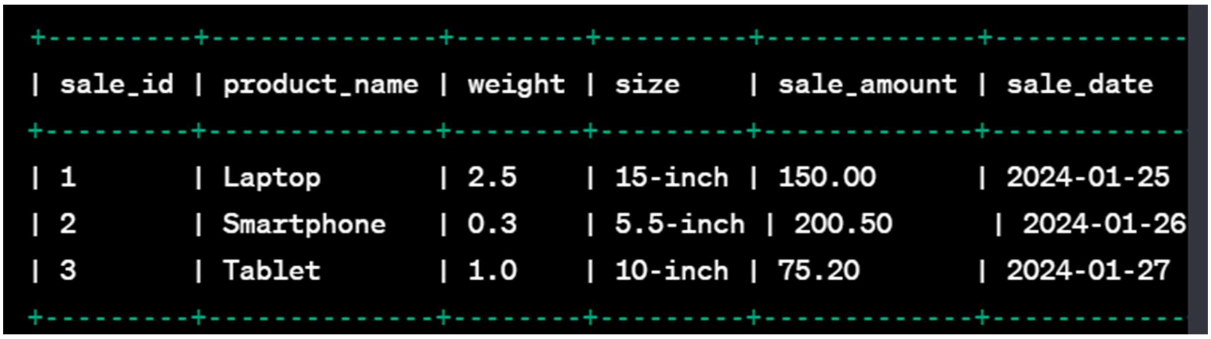
Example query:

```
-- Create Product Table
CREATE TABLE product (
    product_id INT PRIMARY KEY,
    product_name VARCHAR(255)
);

-- Insert sample data into product table
INSERT INTO product (product_id, product_name)
VALUES
    (101, 'Laptop'),
    (102, 'Smartphone'),
    (103, 'Tablet');

-- Create Product Details Table (Normalized)
CREATE TABLE product_details (
    product_id INT PRIMARY KEY,
    weight DECIMAL(5, 2),
    size VARCHAR(20)
);

-- Insert sample data into product details table
INSERT INTO product_details (product_id, weight, size)
```

```
VALUES
   (101, 2.5, '15-inch'),
   (102, 0.3, '5.5-inch'),
   (103, 1.0, '10-inch');

-- Example Snowflake Query
SELECT
   s.sale_id,
   p.product_name,
   pd.weight,
   pd.size,
   s.sale_amount,
   s.sale_date
FROM
   sales s
JOIN
   product p ON s.product_id = p.product_id
JOIN
   product_details pd ON p.product_id = pd.product_id;
```



```
---------+-------------+--------+---------+------------+-----------+
sale_id | product_name | weight | size     | sale_amount | sale_date  |
---------+-------------+--------+---------+------------+-----------+
1        | Laptop       | 2.5    | 15-inch | 150.00      | 2024-01-25 |
2        | Smartphone   | 0.3    | 5.5-inch | 200.50      | 2024-01-26
3        | Tablet       | 1.0    | 10-inch | 75.20       | 2024-01-27 |
---------+-------------+--------+---------+------------+-----------+
-- Example Star Schema Query
SELECT
    s.sale_id,
    p.product_name,
    pd.weight,
    pd.size,
    s.sale_amount,
    s.sale_date
FROM
    sales s
JOIN
    product p ON s.product_id = p.product_id
JOIN
    product_details pd ON p.product_id = pd.product_id;
```



```
+---------+--------------+--------+---------+------------+-----------
| sale_id | product_name | weight | size    | sale_amount | sale_date
+---------+--------------+--------+---------+------------+-----------
| 1       | Laptop       | 2.5    | 15-inch | 150.00      | 2024-01-25
| 2       | Smartphone   | 0.3    | 5.5-inch | 200.50      | 2024-01-26
| 3       | Tablet       | 1.0    | 10-inch | 75.20       | 2024-01-27
+---------+--------------+--------+---------+------------+-----------
```