Basic SQL Querying Courses
Course 2: Analytics

Using Facts and Dimensions data with SnowSQL

Data:	**factsanddimensions.co.uk**
Database Architecture:	**snowflake.com**
Course Design:	**keyanalytics.co.uk**
Supported by:	**analystx.uk**

Contact details on last slides

v20251022_1440

Agenda

1. 1000 to 1010 - Introductions
2. 1010 to 1020 - Run through what we will cover
3. 1020 to 1030 - Overview of Lose_Weight_With_Data - the dataset we will be using
4. 1030 to 1040 - Break
5. 1040 to 1250 - The actual course with a 5min break if needed
6. 1250 to 1350 - Lunch break
7. 1350 to 1400 - Explain how the test works and a few more tips & Q&A.
8. 1400 to 1500 - Test
9. 1500 to 1700 - Extra time if needed

Prerequisites section

1. A computer connected to the internet with a browser (eg Chrome, Firefox)
2. Access to your emails.
3. A Snowflake account (req a free trial here: https://signup.snowflake.com/) See next page for steps.
4. Access to the Facts and Dimensions data. Request via the Snowflake marketplace or contact FAD (https://factsanddimensions.co.uk/contact)
5. Two screens. You could plug a HDMI lead into your TV. Or you can join the meeting again via a 2nd computer or your phone. Just so you can see my shared screen as well as your own.

KeyAnalytics Unlock your Data

FACTS AND DIMENSIONS

snowflake

ANALYST X

Get Connected

If you already have a Snowflake account, skip to step 2

1. Sign up for a free trial Snowflake account
    1. https://signup.snowflake.com/
    2. Fill in your details. Click Continue.
    3. Snowflake edition
        1. Choose Standard edition
        2. Choose Google (the data is available in all 3 regions, but the primary copy is in Google)
        3. From the dropdown, choose London
        4. Tick the Snowflake terms box
        5. Click Get Started
    4. Then you will be asked a few survey questions. You can answer them or click skip.
    5. At the end you will be told you will get an activation email.
    6. Click "Click To Activate"
    7. Create a username (letters and numbers only) and a password. Click Get started.
    8. Once logged in, bookmark the URL. Very important!
2. Once in you will have a "Welcome to Snowflake" splash screen. You might have to zoom out (Ctrl and -) to see the Skip For Now button, bottom-right. Click that.
3. Go to Data Products - Marketplace
4. Search for: Facts and Dimensions Ltd
5. Select one of 2 products:
    1. "Facts and Dimensions All Data Trial + PAYG". Instantly approved access. Free for 30 days.
    2. "Facts and Dimensions All Data". This one includes non open data (eg SNOMED). These datasets need addl data owner T&Cs agreed. If you select this, it will send us an email introducing eachother and we can get that sorted straight away.
6. Click Get or Try for free
7. Click Query Data

KeyAnalytics Unlock your Data

FACTS AND DIMENSIONS

snowflake®

ANALYST X

Course Content

- Brief lecture explaining Facts and Dimension tables
- Quick recap of joins
- Data Granularity
- Main course
  - 1. The basics
    - COUNT, ORDER BY, COUNT, COUNT(DISTINCT), MIN, MAX, AVG, DATEDIFF, <=, >=, >, <, TOP, GROUP BY, BETWEEN, Subquery, WHERE, LEFT (and RIGHT)
  - Break
  - 2. Combine these functions into a single powerful query + HAVING
- Lunch
- Test

KeyAnalytics Unlock your Data

FACTS AND DIMENSIONS

snowflake®

ANALYST X

1. The Basics

Learning Outcomes:

- ORDER BY
- COUNT
- COUNT(DISTINCT)
- MIN
- MAX
- AVG
- DATEDIFF
- DATEADD
- <=, >=, >, <
- TOP
- GROUP BY
- BETWEEN
- WHERE
- LEFT (and RIGHT)
- Date vs next date comparison
- Subquery

KeyAnalytics Unlock your Data | FACTS AND DIMENSIONS | snowflake® | ANALYST X

1.1 Granularity

Learning Outcome:

- Granularity.

What is the granularity of each of these tables?

```
"Lose_Weight_With_Data"."fact_Diet_And_Exercise_Daily_Summary_Data"
"Lose_Weight_With_Data"."fact_Digital_Scales_Data"
"Lose_Weight_With_Data"."fact_Goal_Setting_Data"
```

KeyAnalytics Unlock your Data | FACTS AND DIMENSIONS | snowflake | ANALYST X

1.1 Granularity (Answer)

What is the granularity of each of these tables?

```
"Lose_Weight_With_Data"."fact_Diet_And_Exercise_Daily_Summary_Data
 - Daily. This is my diet and exercise diary where I recorded everything per day.

"Lose_Weight_With_Data"."fact_Digital_Scales_Data"
 - Snapshot. Every time I stood on the scales is a snapshot of my weight.

"Lose_Weight_With_Data"."fact_Goal_Setting_Data"
 - Weekly. This is the weight loss goal per week Sun-Sat.
```

1.2 COUNT

Learning Outcome:

- COUNT
- COUNT(DISTINCT).

1. How many times did I stand on the scales?
2. How many days did I use the scales on?
3. How many diet&exercise diary entries are there?

1.2 COUNT (Answer)

```
1. SELECT COUNT(*) FROM "Lose_Weight_With_Data"."fact_Digital_Scales_Data"
2. SELECT COUNT(DISTINCT LEFT("Time_Of_Measurement",10)) FROM "Lose_Weight_With_Data"."fact_Digital_Scales_Data"
3. SELECT COUNT(DISTINCT "Effective_Snapshot_Date") FROM "Lose_Weight_With_Data"."fact_Diet_And_Exercise_Daily_Summary_Data"
```

KeyAnalytics Unlock your Data | FACTS AND DIMENSIONS | snowflake | ANALYST X

1.3 MIN(), MAX()

Learning Outcome:

- MIN(), MAX()

1. What is the min and max dates with weight snapshots?
2. What was my lightest and heaviest weights?

1.3 MIN, MAX (Answer)

```
1. SELECT MIN("Time_Of_Measurement"),MAX("Time_Of_Measurement")
   FROM "Lose_Weight_With_Data"."fact_Digital_Scales_Data"

2. SELECT MIN("Weight_In_Pounds"),MAX("Weight_In_Pounds")
   FROM "Lose_Weight_With_Data"."fact_Digital_Scales_Data"
```

KeyAnalytics
Unlock your Data

FACTS AND DIMENSIONS

snowflake®

ANALYST X

1.4 DATEDIFF

Learning Outcome:

- DATEDIFF()

1. How many days between when I first stood on the scales and last did?
2. How many days between when I first diarised my diet and exercise and last did?

1.4 DATEDIFF() (Answer)

```
1. SELECT DATEDIFF(d,MIN(LEFT("Time_Of_Measurement",10)),MAX(LEFT("Time_Of_Measurement",10)))
   FROM "Lose_Weight_With_Data"."fact_Digital_Scales_Data"
2. SELECT DATEDIFF(d,MIN("Effective_Snapshot_Date"),MAX("Effective_Snapshot_Date"))
   FROM "Lose_Weight_With_Data"."fact_Diet_And_Exercise_Daily_Summary_Data"
```

1.5 WHERE, <=, >=

Learning Outcome:

- WHERE
- <=
- >=

1. My goal was 150lb. I've been recording my weight long before I started tracking calories and ever since I hit that target. So, there is a range of dates when my proper dieting started and ended.

2. I started at around 210lb and finished when I hit 150lb.

3. When was the last date that I was at least 210lb and when did I first reach 150lb?

1.5 WHERE , <=, >= (Answer)

```
1. SELECT MAX(LEFT("Time_Of_Measurement",10))
   FROM "Lose_Weight_With_Data"."fact_Digital_Scales_Data"
   WHERE "Weight_In_Pounds">=210

2. SELECT MIN(LEFT("Time_Of_Measurement",10))
   FROM "Lose_Weight_With_Data"."fact_Digital_Scales_Data"
   WHERE "Weight_In_Pounds"<=150
```

1.6 TOP 1 with ORDER BY

Learning Outcome:

- TOP 1 with ORDER BY to uncover a record

1. Return the record when I was at my heaviest
2. Return the record when I was at my lightest

1.6 TOP 1 with ORDER BY (Answer)

```
1. SELECT TOP 1 *
   FROM "Lose_Weight_With_Data"."fact_Digital_Scales_Data"
   ORDER BY "Weight_In_Pounds" DESC

2. SELECT TOP 1 *
   FROM "Lose_Weight_With_Data"."fact_Digital_Scales_Data"
   ORDER BY "Weight_In_Pounds" ASC
```

Note: What if there were two different dates when I was joint heaviest? That's where RANK() comes in.

That's for more advanced course with Key Analytics

2. Combine

Learning Outcome:

- HAVING
- Combine use of the basic functions from "Part 1 The Basics" into one powerful query to work out:

The average weight loss per day

2.1 Order By

Learning Outcome:

- View all data in date order

1. Return all weight measurements in date order, where the date range is between when I was last 210lb and first 150lb. Use the dates you uncovered in previous section.

Note: normally we would use variables for storing this; that, and a LOT more besides, is covered in more advanced SQL courses by Key Analytics.

2.1 Order By (Answer)

```
SELECT LEFT("Time_Of_Measurement",10),"Weight_In_Pounds"
FROM "Lose_Weight_With_Data"."fact_Digital_Scales_Data"
WHERE LEFT("Time_Of_Measurement",10) BETWEEN '2023-09-27' AND '2024-12-13'
ORDER BY LEFT("Time_Of_Measurement",10);
```

2.2 Aggregate using GROUP BY

Learning Outcome:

- Get the min (or max or avg etc) value of one metric for each value in one dimension

1. I stood on the scales several times of the day. We just want one measurement per day. Update your code to return lowest weight of the day in date order.

2.2 Aggregate (min) using GROUP BY (Answer)

```
SELECT LEFT("Time_Of_Measurement",10),MIN("Weight_In_Pounds")
FROM "Lose_Weight_With_Data"."fact_Digital_Scales_Data"
WHERE LEFT("Time_Of_Measurement",10) BETWEEN '2023-09-27' AND '2024-12-13'
GROUP BY LEFT("Time_Of_Measurement",10)
ORDER BY LEFT("Time_Of_Measurement",10);
```

KeyAnalytics Unlock your Data FACTS AND DIMENSIONS snowflake® ANALYST X

2.3 Join different dates with DATEADD

Learning Outcome:

- Join a table back on itself on different dates using DATEADD()

1. Get the change in weight per day by joining the scales data table back on itself each day = each next day

2.3 Join different dates with DATEADD (Answer)

```
SELECT
            min(a."Weight_In_Pounds") as "MyBestWeightOfTheDay_Today"
            ,min(b."Weight_In_Pounds") as "MyBestWeightOfTheDay_Yesterday"
            ,min(a."Weight_In_Pounds") - min(b."Weight_In_Pounds") as "WeightLossSinceYesterday"
            ,LEFT(a."Time_Of_Measurement",10) as "TodaysDate"
FROM
            "Lose_Weight_With_Data"."fact_Digital_Scales_Data" a INNER JOIN
            "Lose_Weight_With_Data"."fact_Digital_Scales_Data" b ON
             LEFT(a."Time_Of_Measurement",10) = DATEADD(d,1,b."Effective_Snapshot_Date")
WHERE
            LEFT(a."Time_Of_Measurement",10) BETWEEN '2023-09-27' AND '2024-12-13'
GROUP BY
            LEFT(a."Effective_Snapshot_Date",10)
ORDER BY
            LEFT(a."Effective_Snapshot_Date",10);
```

2.4 Subquery

Learning Outcome:

- Embed one select query inside another.

1. Get the avg, min, max change in weight over the period

2.4 Subquery (Answer)

```
SELECT
    AVG("WeightLossSinceYesterday")
    ,MIN("WeightLossSinceYesterday")
    ,MAX("WeightLossSinceYesterday")
FROM
(SELECT
            min(a."Weight_In_Pounds") as "MyBestWeightOfTheDay_Today"
            ,min(b."Weight_In_Pounds") as "MyBestWeightOfTheDay_Yesterday"
            ,min(a."Weight_In_Pounds") - min(b."Weight_In_Pounds") as "WeightLossSinceYesterday"
            ,LEFT(a."Effective_Snapshot_Date",10) as "TodaysDate"
FROM
            "Lose_Weight_With_Data"."fact_Digital_Scales_Data" a INNER JOIN
            "Lose_Weight_With_Data"."fact_Digital_Scales_Data" b ON
            LEFT(a."Effective_Snapshot_Date",10) = DATEADD(d,1,b."Effective_Snapshot_Date")
WHERE
            LEFT(a."Effective_Snapshot_Date",10) BETWEEN '2023-09-27' AND '2024-12-13'
GROUP BY
            LEFT(a."Effective_Snapshot_Date",10)
ORDER BY
             LEFT(a."Effective_Snapshot_Date",10))a;
```

2.5 HAVING

Learning Outcome:

- Use HAVING to filter results of an aggregate query

1. Use HAVING to work out my avg weight loss when I was >200lb
2. Change it to see how it looks when <=200lb

2.2 HAVING (Answer: over 200lb)

```
SELECT
    AVG("WeightLossSinceYesterday")
    ,MIN("WeightLossSinceYesterday")
    ,MAX("WeightLossSinceYesterday")
FROM
(SELECT
            MIN(a."Weight_In_Pounds") AS "MyBestWeightOfTheDay_Today"
            ,MIN(b."Weight_In_Pounds") AS "MyBestWeightOfTheDay_Yesterday"
            ,MIN(a."Weight_In_Pounds") - MIN(b."Weight_In_Pounds") AS "WeightLossSinceYesterday"
            ,LEFT(a."Effective_Snapshot_Date",10) AS "TodaysDate"
FROM
            "Lose_Weight_With_Data"."fact_Digital_Scales_Data" a INNER JOIN
            "Lose_Weight_With_Data"."fact_Digital_Scales_Data" b ON
            LEFT(a."Effective_Snapshot_Date",10) = DATEADD(d,1,b."Effective_Snapshot_Date")
WHERE
            LEFT(a."Effective_Snapshot_Date",10) BETWEEN '2023-09-27' AND '2024-12-13'
GROUP BY
            LEFT(a."Effective_Snapshot_Date",10)
HAVING
            MIN(a."Weight_In_Pounds") > 200
ORDER BY
            LEFT(a."Effective_Snapshot_Date",10))a;
```

2.2 HAVING (Answer: <=200lb)

```
SELECT
    AVG("WeightLossSinceYesterday")
    ,MIN("WeightLossSinceYesterday")
    ,MAX("WeightLossSinceYesterday")
FROM
(SELECT
            MIN(a."Weight_In_Pounds") AS "MyBestWeightOfTheDay_Today"
            ,MIN(b."Weight_In_Pounds") AS "MyBestWeightOfTheDay_Yesterday"
            ,MIN(a."Weight_In_Pounds") - MIN(b."Weight_In_Pounds") AS "WeightLossSinceYesterday"
            ,LEFT(a."Effective_Snapshot_Date",10) AS "TodaysDate"
FROM
            "Lose_Weight_With_Data"."fact_Digital_Scales_Data" a INNER JOIN
            "Lose_Weight_With_Data"."fact_Digital_Scales_Data" b ON
            LEFT(a."Effective_Snapshot_Date",10) = DATEADD(d,1,b."Effective_Snapshot_Date")
WHERE
            LEFT(a."Effective_Snapshot_Date",10) BETWEEN '2023-09-27' AND '2024-12-13'
GROUP BY
            LEFT(a."Effective_Snapshot_Date",10)
HAVING
            MIN(a."Weight_In_Pounds") <= 200
ORDER BY
            LEFT(a."Effective_Snapshot_Date",10))a;
```

FACTS AND DIMENSIONS

Filipe McManus

CEO

Facts and Dimensions Ltd

Website: https://factsanddimensions.co.uk/

Linkedin: https://www.linkedin.com/company/factsanddimensions

Youtube: https://www.youtube.com/@factsanddimensions/playlists/

Janet Broome

Account Director NHS

Snowflake

Website: https://www.snowflake.com//

Linkedin: https://www.linkedin.com/company/snowflake-computing/

Youtube: https://www.youtube.com/@snowflakedevelopers/

KeyAnalytics
Unlock your Data

Kay Khan

CEO

Key Analytics

Website: https://www.keyanalytics.co.uk

Linkedin: https://www.linkedin.com/company/keyanalytics/

Alex Cheung

Senior Manager

NHS AnalystX

Website: https://analystx.uk/

Linkedin: https://www.linkedin.com/company/analystx/