



Basic SQL Querying Courses

Course 1: Joins

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

v20241106_1527

Agenda

1. 0930 to 0940 - Introductions and quick run through what we will cover
2. 0940 to 1020 - Lecture explaining fundamentals of joins.
3. 1020 to 1030 - Explain importance of understanding the source data
4. 1030 to 1040 - 10 min tea break
5. 1040 to 1250 - The actual course with 5min break
6. 1250 to 1340 - Lunch break
7. 1340 to 1350 - Explain how the test works and a few more tips & Q&A.
8. 1350 to 1450 - Test
9. 1450 to 1500 - Finish up
10. 1500 to 1520 - Extra time if needed



KeyAnalytics
Unlock your Data



FACTS AND
DIMENSIONS



snowflake®

ANALYST



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



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



Course Content

- Brief lecture explaining Facts and Dimension tables
- PART 1
 - SELECT...FROM, Alias
 - LEFT JOIN, INNER JOIN
 - IS NULL, IFNULL
 - DISTINCT
 - CASE statement in FROM section
 - CASE statement in SELECT section
 - ORDER BY
- PART 2 (Handling duplicates)
 - Filtering dimension table to remove duplicates
 - Ensuring your filtered left join does not act like an inner join
- FINAL TEST



KeyAnalytics
Unlock your Data



**FACTS AND
DIMENSIONS**



snowflake®

ANALYST



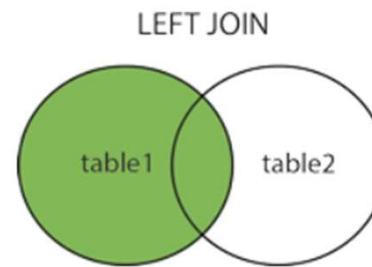
What is a “Join” in SQL?

- If two or multiple tables have a **common** column in them, a **join** clause can be used to combine data from the tables
- Allows us to see data from different tables based on a relationship between them
- Main methods of joining tables:
 - ✓ Inner Join
 - ✓ Left Join
 - ✓ Right Join (not covered in this session)
 - ✓ Full Join (not covered in this session)

Left Join

- A left join will combine two tables based on a **common column(s)**
- This type of join will give **all** rows from the left table (table1) and **matching** rows from the right table (table2)
- Left Join will give **all** rows from left table even if there are no matches in the right table
- Important: If your where clause filters on table on right, that makes it an inner join

```
SELECT column1, column2, column3 ...  
FROM table1 Left table  
LEFT JOIN table2 Right table  
ON table1.column= table2.column
```



Facts And Dimensions

- Fact Table: A table of data such as population counts, A&E activity, waiting list numbers
- Dimension Table: A table that describes a code.



KeyAnalytics
Unlock your Data



FACTS AND
DIMENSIONS



snowflake®

ANALYST



PART 1. No Duplicates Starting Code

Copy this code into a new Worksheet in Snowflake:

```
select
  a."Organisation_Code"
,a."Staff_Group"
,a."FTE_Days_Available"
,a."FTE_Days_Lost_All"
,a."FTE_Days_Lost_COVID"
,a."Effective_Snapshot_Date"
,a."Report_Period_Length"
from
  "NHS_Workforce"."fact_Sickness_Absence_COVID19_Related" a
```

Make a note of how many rows you get.
Why would we do this before building rest of the code?
Also, why did I use an alias given there is only one table?



KeyAnalytics
Unlock your Data



**FACTS AND
DIMENSIONS**



snowflake®

ANALYST



Challenge 1.1

Learning Outcome:

- Join a fact table to a dimension table to return the descriptions for some codes.
- See what you get if a code in the fact table does not exist in the dimension table.

Challenge

Firstly, keep track of the row count in fact table to use as a check.

Select data from **"NHS_Workforce".**
fact_Sickness_Absence_COVID19_Related" and
include an additional column from
"ODS"."dim_All_Codes" showing the organisation
names. Use a left join to link both tables.

Challenge 1.1: Answer

```
select
  a."Organisation_Code"
  ,b."Name" as "Organisation_Name"
  ,a."Staff_Group"
  ,a."FTE_Days_Available"
  ,a."FTE_Days_Lost_All"
  ,a."FTE_Days_Lost_COVID"
  ,a."Effective_Snapshot_Date"
  ,a."Report_Period_Length"
from
  "NHS_Workforce"."fact_Sickness_Absence_COVID19_Related" a
left join "ODS"."dim_All_Codes" b on
  a."Organisation_Code" = b."Code"
```

How many rows do you get now?
Did any rows return Organisation_Name as NULL?



KeyAnalytics
Unlock your Data



FACTS AND
DIMENSIONS



snowflake®

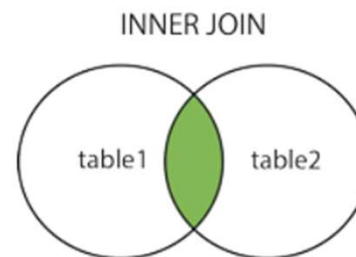
ANALYST



Inner Join

- Selects **common rows** between the two tables **with the matching field** (matching column that is present in both tables)
- SQL returns a result set that includes all the columns in the 'select' statement as well as the matching field
- The primary key in one table will be the foreign key in the other table
- If there are rows in table1 that **don't have** matches in the table2, these rows **won't appear** in the result set

```
SELECT column1, column2, column3 ...  
FROM table1  
INNER JOIN table2  
ON table1.column1 = table2.column
```



Challenge 1.2

Learning Outcome:

- See what happens when you join two tables and only return rows where a match exists.

Challenge

Change your join to an INNER JOIN



KeyAnalytics
Unlock your Data



FACTS AND
DIMENSIONS



snowflake®

ANALYST



Challenge 1.2: Answer

```
select
  a."Organisation_Code"
,b."Name" as "Organisation_Name"
,a."Staff_Group"
,a."FTE_Days_Available"
,a."FTE_Days_Lost_All"
,a."FTE_Days_Lost_COVID"
,a."Effective_Snapshot_Date"
,a."Report_Period_Length"
from
  "NHS_Workforce"."fact_Sickness_Absence_COVID19_Related" a
  inner join ODS."dim_All_Codes" b on
    a."Organisation_Code" = b."Code"
```

How many rows do you get?

What do you notice about rows where Organisation_Code = 'All organisations'?

Do you have any rows with no Organisation_Name?

Data quality issues

- Left, right or full outer joins can return NULLS which may not be expected
- If you only want rows in left table that have a match, doing an inner join can solve this as it will return only rows that.

Challenge 1.3

Learning Outcome:

- Uncovering codes with no record in the foreign table.

Challenge:

Select data from **"NHS_Workforce".**
fact_Sickness_Absence_COVID19_Related" but only
those where no code in **"ODS"."dim_All_Codes"**
exists.

Challenge 1.3: Answer

```
select
  a."Organisation_Code"
  ,b."Name" as "Organisation_Name"
  ,a."Staff_Group"
  ,a."FTE_Days_Available"
  ,a."FTE_Days_Lost_All"
  ,a."FTE_Days_Lost_COVID"
  ,a."Effective_Snapshot_Date"
  ,a."Report_Period_Length"
from
  "NHS_Workforce"."fact_Sickness_Absence_COVID19_Related" a
left join ODS."dim_All_Codes" b on
  a."Organisation_Code" = b."Code"
where
  b."Name" is null
```

How many rows do you get?

How did it return only the rows you see?

Challenge 1.4

Learning Outcome:

- Using Distinct

Challenge

Too many rows. You just want the distinct list of problem codes.
Return a distinct list of the org codes that have no found a match.

Challenge 1.4: Answer

```
select distinct
  a."Organisation_Code"
  ,b."Name" as "Organisation_Name"
  --, a."Staff_Group"
  --, a."FTE_Days_Available"
  --, a."FTE_Days_Lost_All"
  --, a."FTE_Days_Lost_COVID"
  --, a."Effective_Snapshot_Date"
  --, a."Report_Period_Length"
from
  "NHS_Workforce"."fact_Sickness_Absence_COVID19_Related" a left join
  "ODS"."dim_All_Codes" b on
  a."Organisation_Code" = b."Code"
where
  b."Name" is null
```

Challenge 1.5

Learning Outcome:

- Using a CASE statement in a join.
- See: <https://docs.snowflake.com/en/sql-reference/functions/case>

Challenge

Some codes in the fact table end in “-X”. They are not in the dim table; but are with the “-X” removed.
Update the code to remove the “-X” from the fact table codes.

To identify the records ending in “-X”, add an additional where clause:

and a."Organisation_Code" like '%-X'

Alter your join by using a CASE statement to take account of those -Xs

Challenge 1.5: Answer

```
select distinct
  a."Organisation_Code"
,b."Name" as "Organisation_Name"
--, a."Staff_Group"
--, a."FTE_Days_Available"
--, a."FTE_Days_Lost_All"
--, a."FTE_Days_Lost_COVID"
--, a."Effective_Snapshot_Date"
--, a."Report_Period_Length"
from
  "NHS_Workforce"."fact_Sickness_Absence_COVID19_Related" a left join
  "ODS"."dim_All_Codes" b on
  case
    when a."Organisation_Code" like '____-X'
    then left(a."Organisation_Code",3)
    else a."Organisation_Code"
  end = b."Code"
where
  b."Name" is null
```

Challenge 1.6

Learning Outcome:

- Add a name description where there isn't a match in the join.
- IS NULL

Challenge

We have just one last org code with no name: "All organisations"

For "Organisation_Name", you are currently returning b."Name". The problem is, there is no match for the code 'All organisations', so replace b."Name" with a case statement that will return 'All organisations' if a."Organisation_Name" = 'All organisations' else return b."Name".

Challenge 1.6:

Answer

```
select distinct
  a."Organisation_Code"
,case
  when a."Organisation_Code" = 'All organisations'
  then 'All orgs'
  else b."Name"
end as "Organisation_Name"
--, a."Staff_Group"
--, a."FTE_Days_Available"
--, a."FTE_Days_Lost_All"
--, a."FTE_Days_Lost_COVID"
--, a."Effective_Snapshot_Date"
--, a."Report_Period_Length"
from
  "NHS_Workforce"."fact_Sickness_Absence_COVID19_Related" a left join
  "ODS"."dim_All_Codes" b on
  case
    when a."Organisation_Code" like '____-X'
    then left(a."Organisation_Code",3)
    else a."Organisation_Code"
  end = b."Code"
where
  b."Name" is null
```



KeyAnalytics
Unlock your Data



FACTS AND
DIMENSIONS



snowflake®

ANALYST



Challenge 1.7

Learning Outcome:

- Use ORDER BY to help eye-balling results

Challenge

Remove your where clause and let's see the codes and their names.

Use ORDER BY to see all codes in order, so we can easily scroll down to find codes 'All organisations' and 'RYK-X' and also look at other codes too.



KeyAnalytics
Unlock your Data



**FACTS AND
DIMENSIONS**



snowflake®



ANALYST

Challenge 1.7:

Answer

```
select distinct
  a."Organisation_Code"
,case
  when b."Name" is null and a."Organisation_Code" = 'All organisations'
  then 'All organisations'
  else b."Name"
end as "Organisation_Name"
--, a."Staff_Group"
--, a."FTE_Days_Available"
--, a."FTE_Days_Lost_All"
--, a."FTE_Days_Lost_COVID"
--, a."Effective_Snapshot_Date"
--, a."Report_Period_Length"
from
  "NHS_Workforce"."fact_Sickness_Absence_COVID19_Related" a left join
  "ODS"."dim_All_Codes" b on
  case
    when a."Organisation_Code" like '____-X'
    then left(a."Organisation_Code",3)
    else a."Organisation_Code"
  end = b."Code"
order by
  a."Organisation_Code"
```



KeyAnalytics
Unlock your Data



FACTS AND
DIMENSIONS



snowflake®



Challenge 1.8

Learning Outcome:

- Check final row count

Challenge

Uncomment all the columns so all columns are returned

Remove the DISTINCT so all rows are returned

Remove ORDER BY as no longer needed

Challenge 1.8:

Answer

```
select
  a."Organisation_Code"
,case
  when a."Organisation_Code" = 'All organisations'
  then 'All orgs'
  else b."Name"
end as "Organisation_Name"
, a."Staff_Group"
, a."FTE_Days_Available"
, a."FTE_Days_Lost_All"
, a."FTE_Days_Lost_COVID"
, a."Effective_Snapshot_Date"
, a."Report_Period_Length"
from
  "NHS_Workforce"."fact_Sickness_Absence_COVID19_Related" a left join
  "ODS"."dim_All_Codes" b on
  case
    when a."Organisation_Code" like '____-X'
    then left(a."Organisation_Code",3)
    else a."Organisation_Code"
  end = b."Code"
```

PART 2: Handling Duplicates Starter Code

```
select
  a."Organisation_Code"
  ,b."Name" as "Organisation_Name"
  ,a."Staff_Group"
  ,a."FTE_Days_Available"
  ,a."FTE_Days_Lost_All"
  ,a."FTE_Days_Lost_COVID"
  ,a."Effective_Snapshot_Date"
  ,a."Report_Period_Length"
from
  "NHS_Workforce"."fact_Sickness_Absence_COVID19_Related" a
left join "ODS_API"."dim_Organisation_SCD" b on a."Organisation_Code" = b."OrgId_extension"
```

We will now switch to a different dimension table so we can learn about duplicates.

Replace your code with the above. We are now using table ODS_API."dim_Organisation_SCD"

Unwanted duplicates

- Already having duplicates in table(s)
- Joining tables that already have duplicates e.g. a patient dim has rows with unique ID's and a country dim has rows without unique ID's

Patient_ID	First_Name	Age	Location_ID
1	Natalie	20	1
2	Bob	30	2
3	Sophie	22	2

Location_ID	Location_Name
1	England
1	England
2	Wales

Patient_ID	First_Name	Age	Location_Name
1	Natalie	20	England
1	Natalie	20	England
2	Bob	30	Wales
3	Sophie	22	Wales

Challenge 2.1

Learning Outcome:

- Handling duplicates without losing records.

Challenge

Alter your code so you do not get duplicates



KeyAnalytics
Unlock your Data



FACTS AND
DIMENSIONS



snowflake®

ANALYST



Challenge 2.1: Answer

```
select
  a."Organisation_Code"
  ,b."Name" as "Organisation_Name"
  ,b."PostCode"
  ,b."UPRN"
  ,a."Staff_Group"
  ,a."FTE_Days_Available"
  ,a."FTE_Days_Lost_All"
  ,a."FTE_Days_Lost_COVID"
  ,a."Effective_Snapshot_Date"
  ,a."Report_Period_Length"
from
  "NHS_Workforce"."fact_Sickness_Absence_COVID19_Related" a
  left join "ODS_API"."dim_Organisation_SCD" b on
    a."Organisation_Code" = b."OrgId_extension"
where
  b."Is_Latest" = 1
```

How many rows do you get?

Given that ODS_API."dim_Organisation_SCD" has several rows per Organisation Code, how did you prevent returning duplicates?

Challenge 2.2

Learning Outcome:

- Write a where clause that does not cause your LEFT JOIN to act like an INNER JOIN.

Challenge

Now you're missing records though. Why is that? Alter the code so we get the all fact table rows without duplicates and without missing any.



KeyAnalytics
Unlock your Data



FACTS AND
DIMENSIONS



snowflake®

ANALYST



Challenge 2.2: Answer

```
select
  a."Organisation_Code"
  ,b."Name" as "Organisation_Name"
  ,b."PostCode"
  ,b."UPRN"
  ,a."Staff_Group"
  ,a."FTE_Days_Available"
  ,a."FTE_Days_Lost_All"
  ,a."FTE_Days_Lost_COVID"
  ,a."Effective_Snapshot_Date"
  ,a."Report_Period_Length"
from
  "NHS_Workforce"."fact_Sickness_Absence_COVID19_Related" a
left join "ODS_API"."dim_Organisation_SCD" b on
  a."Organisation_Code" = b."OrgId_extension"
Where
  b."Is_Latest" = 1 or b."Is_Latest" is null
```

The correct number of rows is returned. Why is that?
btw you could use IFNULL(b."Is_Latest",1) = 1



KeyAnalytics
Unlock your Data



FACTS AND
DIMENSIONS



snowflake®

ANALYST



Challenge 2.3

Learning Outcome:

- Use LIKE or ILIKE for single character wildcards, using _

Challenge

Return all the rows where Staff_Group has the word manager in it.

Btw, Snowflake is case sensitive. Use ILIKE instead of LIKE.



KeyAnalytics
Unlock your Data



FACTS AND
DIMENSIONS



snowflake®

ANALYST



Challenge 2.3: Answer

```
select
  a."Organisation_Code"
,b."Name" as "Organisation_Name"
,b."PostCode"
,b."UPRN"
,a."Staff_Group"
,a."FTE_Days_Available"
,a."FTE_Days_Lost_All"
,a."FTE_Days_Lost_COVID"
,a."Effective_Snapshot_Date"
,a."Report_Period_Length"
from
  "NHS_Workforce"."fact_Sickness_Absence_COVID19_Related" a
left join "ODS_API"."dim_Organisation_SCD" b on
  a."Organisation_Code" = b."OrgId_extension"
Where
  ifnull("Is_Latest",1) = 1
  and a."Staff_Group" ilike '%manager%'
```

Challenge 2.4

Learning Outcome:

- Use LIKE or ILIKE for multiple wildcard characters, using %

Challenge

Add another clause that only returns ICBs, that is codes that are 3 characters long and start with a Q.



KeyAnalytics
Unlock your Data



FACTS AND
DIMENSIONS



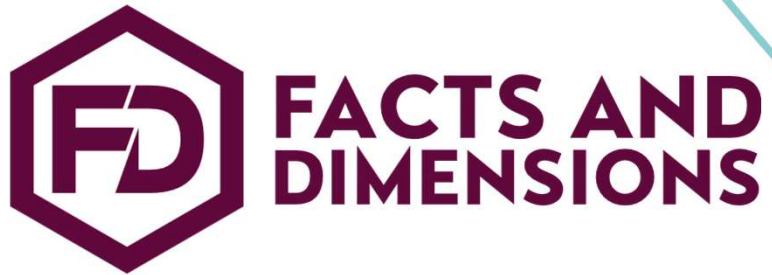
snowflake®

ANALYST



Challenge 2.4: Answer

```
select
  a."Organisation_Code"
,b."Name" as "Organisation_Name"
,b."PostCode"
,b."UPRN"
,a."Staff_Group"
,a."FTE_Days_Available"
,a."FTE_Days_Lost_All"
,a."FTE_Days_Lost_COVID"
,a."Effective_Snapshot_Date"
,a."Report_Period_Length"
from
  "NHS_Workforce"."fact_Sickness_Absence_COVID19_Related" a
left join "ODS_API"."dim_Organisation_SCD" b on
  a."Organisation_Code" = b."OrgId_extension"
Where
  b."Is_Latest" = 1 or b."Is_Latest" is null
and a."Staff_Group" ilike '%manager%'
and "Organisation_Code" like 'Q__'
```



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

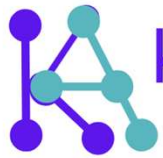
Snowflake



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/>