



# Power BI - Intermediate

Grant Shannon

Oct 2024

,



# Housekeeping

- Student check
- Installation and environment setup
  - Download install PBI desktop client
  - Connect to github repo
    - <https://github.com/gpsuser/PBI>
  - Get data files
- 10 min break on the hour (time permitting)
- Hands on course



# Agenda

- Data Modelling Concepts
  - Dimensional Modelling
- Dimensional Modelling - Business Case
- Designing a Data Model
  - Dimensional Model
- Implementing a Data Model
  - Dimensional Model – Star Schema
- Including Measures in the Data Model
- Dax Aggregation within a Dimensional Model
- Generating Reports



# Data Modelling



# Data Modelling Re-cap

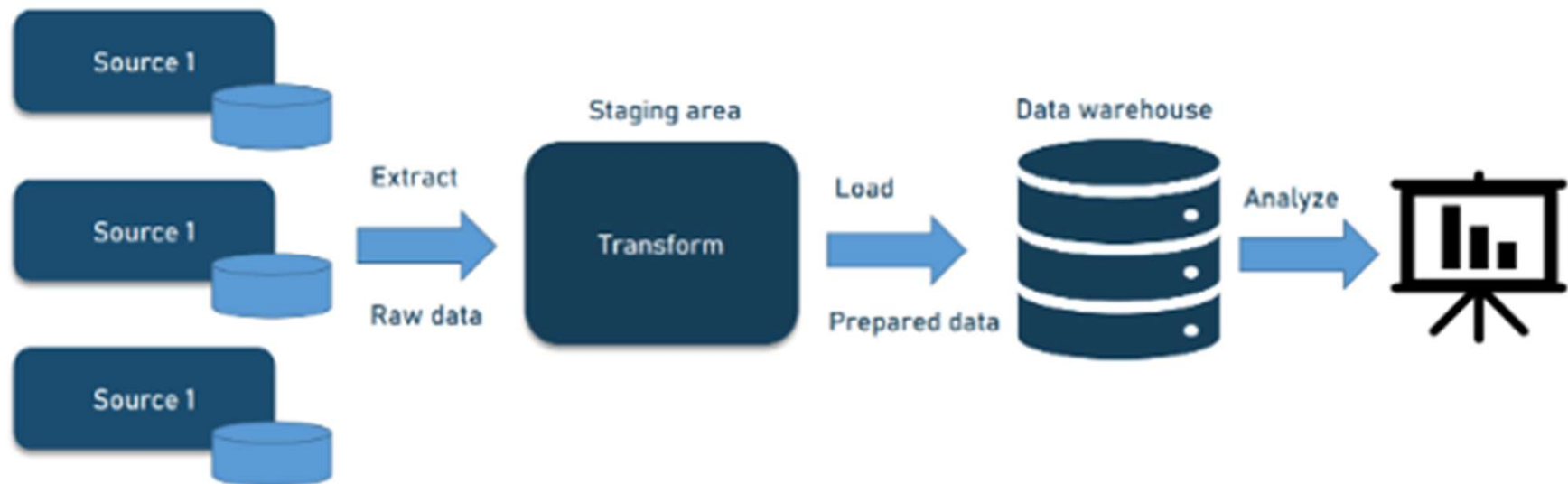
- Data warehouse – large store of data for analytics and reporting
- Data model – structured and logical organization of data showing relationship between data elements
- Multidimensional schema – model different dimensions to track entities/actions concerning data warehouse activities
  - Referred to as dimensional modelling
- A good starting point to understanding multidimensional schema is the star schema

# Business logic behind dimensional modelling

- Dimensional models:
  - optimise performance,
  - simplify data management,
  - enhances analytical capabilities, and
  - supports scalability and flexibility
- Preferred choice for data warehousing and business intelligence applications
  - Dimensional modelling: On-premises vs. cloud implementations

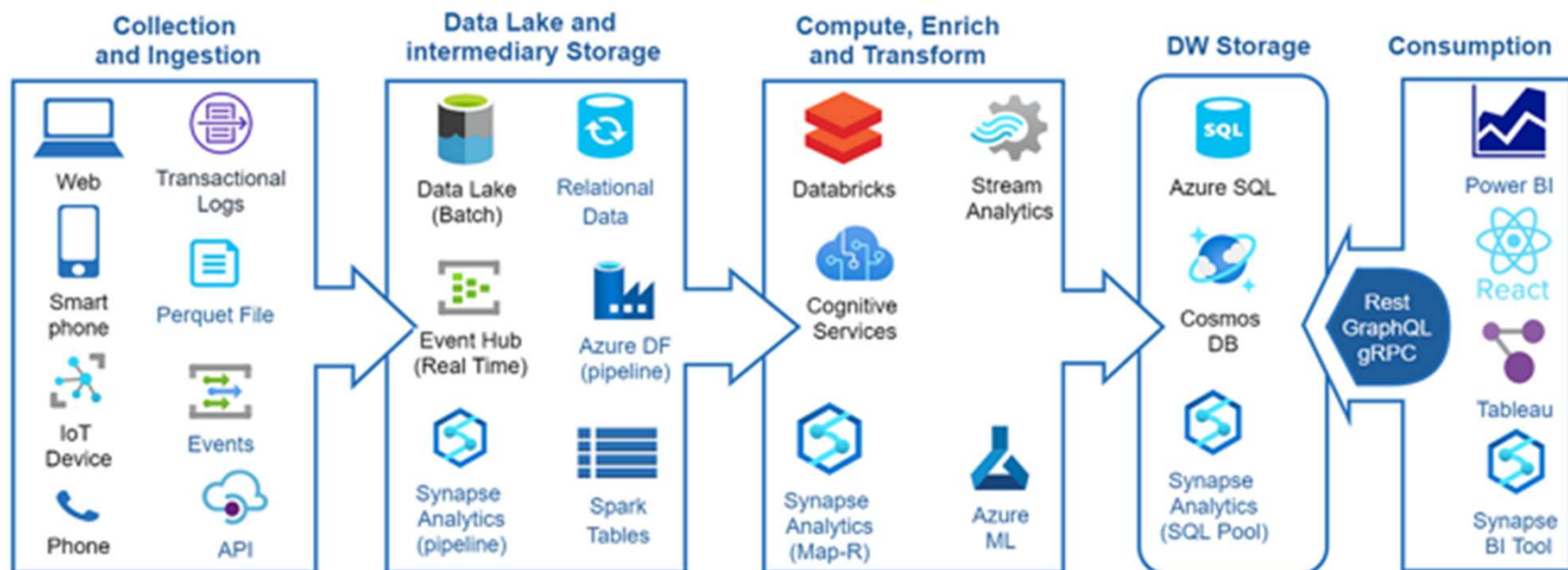
# On Premise Pattern

- On premise Extract Transform Load (ETL) architecture



# Cloud Pattern

- Cloud based Extract Load Transform (ELT) architecture





# Business logic behind dimensional modelling

more detail

- **Improved Query Performance**
- **Fact tables** store large amounts of transactional data, while **dimension tables** store descriptive attributes. This separation allows for more efficient queries, as the database can quickly join smaller dimension tables with the fact table
- **2. Simplified Data Management**
- By organizing data into fact and dimension tables, you reduce redundancy and ensure consistency. Dimension tables centralize descriptive data, making it easier to update and maintain
- **3. Enhanced Data Analysis**
- Star schemas facilitate complex queries and reporting. Analysts can easily slice and dice data along different dimensions (e.g., time, product, customer) to gain insights

# Business logic behind dimensional modelling

more detail

- **4. Scalability**

- Fact tables can grow significantly over time, but dimension tables typically remain relatively small. This structure supports scalability, allowing the data warehouse to handle large volumes of data without performance degradation

- **5. Flexibility**

- Star schemas provide flexibility in data modeling. You can add new dimensions or facts without disrupting existing queries and reports, making it easier to adapt to changing business needs

- **6. Data Integrity**

- Normalizing data into fact and dimension tables helps maintain data integrity. Each piece of information is stored once, reducing the risk of inconsistencies

- **7. User-Friendly**

- Star schemas are intuitive and easier for business users to understand. The clear separation between facts and dimensions aligns well with how users think about their data



# Dimensional Modelling example

- Start with single table of raw data
- Split into single Fact and multiple Dimension tables to create Star Schema

# Start with Single Table

Raw data (denormalised)

SaleID	ProductName	CustomerName	SaleDate	Amount
1	Widget A	Alice	2024-01-01	100.00
2	Widget B	Bob	2024-01-02	150.00
3	Widget C	Charlie	2024-01-03	200.00
4	Widget A	David	2024-01-04	100.00
5	Widget B	Eve	2024-01-05	150.00



# Concepts

## Keys and Normalisation

- Primary Keys (part of fact table and dimension tables)
- Foreign Keys (part of fact table)
- Normalisation
  - Creating separate tables in dimensional architectures aims to reduce redundancy - and the resulting tables are said to be in Normal Form



# Transform into Multiple Tables

- Star Shema
  - Single Fact Table
  - Multiple Dimension Tables
- Tables connect to each other through KEY columns
- Fact Table
  - contains a PRIMARY KEY and FOREIGN KEYS
- Dimension Tables
  - contain PRIMARY KEYS that connect to the Fact table FOREIGN\_KEYS



# Normalisation vs. Denormalisation

- Creating separate tables in dimensional architectures aims to reduce redundancy - and the resulting tables are said to be in Normal Form
  - There are varying degrees of normal form
  - These normal forms help in organizing data efficiently, reducing redundancy, and ensuring data integrity.
- The initial raw data table – before creating fact and dimension tables – is referred to as being Denormalised and contains redundant or repeated data

# Data Modelling – Star Schema

## Fact Table

- Fact table is the central table in a star schema
- Fact tables typically contain measurable information relating to an event
  - such as a sales or transaction event
  - can contain quantitative information
  - can contain string information
  - contains PRIMARY KEY and FOREIGN KEYS
- Fact table connects to Dimension tables with FOREIGN KEYS



# Data Modelling – Star Schema

## Dimension Tables

- Dimension tables hold PRIMARY KEYS that connect to the FOREIGN KEYS in the Fact table
- Dimension tables contain descriptive information
- Example:
  - Dim\_Product table contains descriptive information about products – that are identifiable in the Fact table through the relevant FOREIGN KEY



# Sales Table

## Fact Table

SaleID	ProductID	CustomerID	SaleDate	Amount
1	1	1	2024-01-01	100.00
2	2	2	2024-01-02	150.00
3	3	3	2024-01-03	200.00
4	1	4	2024-01-04	100.00
5	2	5	2024-01-05	150.00

# Products Table

## Dimension Table

ProductID	ProductName
1	Widget A
2	Widget B
3	Widget C

# Customers Table

## Dimension Table

CustomerID	CustomerName
1	Alice
2	Bob
3	Charlie
4	David
5	Eve



# Hands On Lab

- Create a Star Schema in Power BI
- Import the Fact and Dimension tables
- Join tables
- Extend Dimensions
- Create measures
- Create tables with DAX



# Get Data

## Home

- > Get Data > Excel Workbook
- Navigate to project folder
  - > Raw Data > Open> Raw Data> Transform Data > Date Time > Data Type > Date/Time > Replace Current > Close &Apply > Close & Apply

## Home

- > Get Data > Excel Workbook
- Navigate to project folder
  - > Fact Sales > Open> Fact Sales > Transform Data > Date Time > Data Type > Date/Time > Replace Current > Close &Apply > Close & Apply



# Get Data

## Home

- > Get Data > Excel Workbook
- Navigate to project folder
  - > Dim Customer > Open> Dim Customer > Load

## Home

- > Get Data > Excel Workbook
- Navigate to project folder
  - > Dim Product > Open> Dim Product > Load



# Get Data

## Home

- > Get Data > Excel Workbook
- Navigate to project folder
  - > Dim City > Open> Dim City > Load

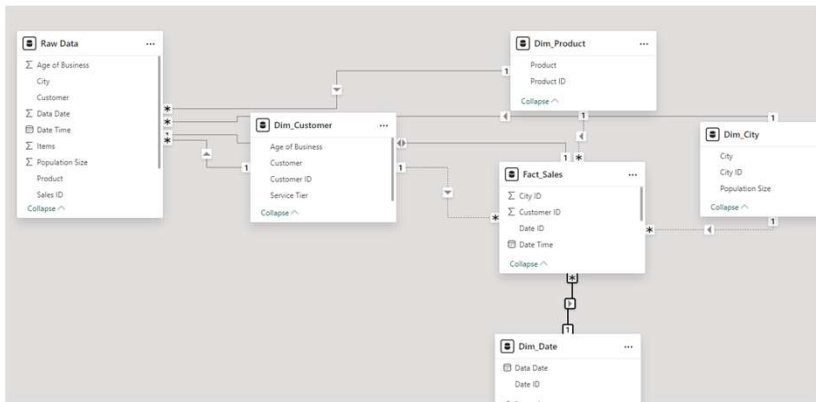
## Home

- > Get Data > Excel Workbook
- Navigate to project folder
  - > Dim Date > Open> Dim Date > Transform Data > Data Date > Data Type > Date > Replace Current > Close &Apply > Close & Apply



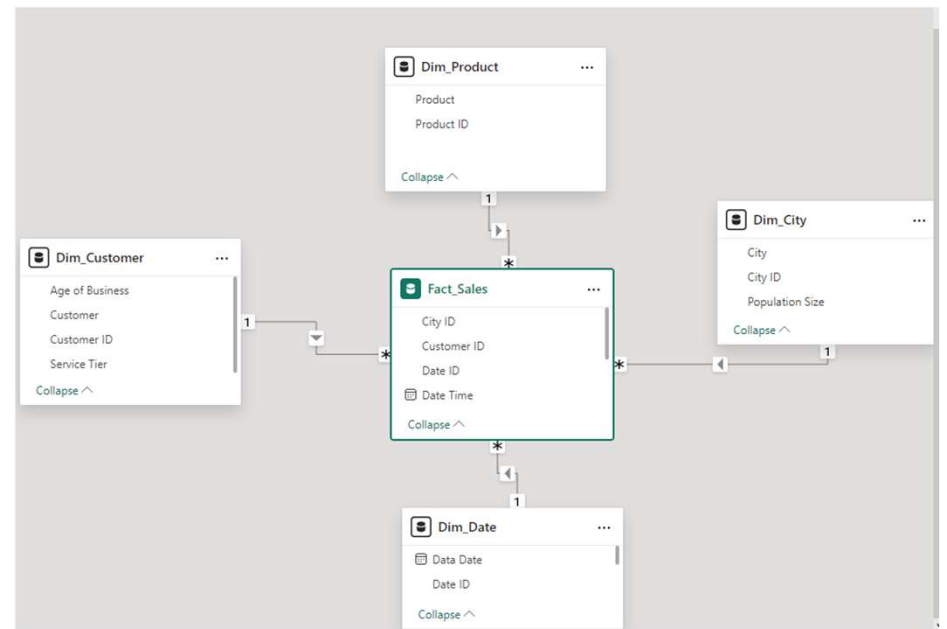
# Disconnect Raw Data Relationships

## Recreate Relationships



Given this

Want this



# Extend Date Dimension Table

## Insert Columns

- Select >
  - Dim\_Date
- New Column >
  - Formatted Date = `FORMAT(Dim_Date[Data Date], "YYYY-mm-dd")`
- New Column >
  - Year = `YEAR(Dim_Date[Data Date])`
- New Column >
  - Month = `INT(RIGHT(LEFT(Dim_Date[Formatted Date], 7), 2))`
- New Column >
  - Quarter = `QUARTER(Dim_Date[Data Date])`

# Extended Date Dimension Table

File Home Help External tools **Table tools**

Name Dim\_Date

Mark as date table  
Calendars

Manage relationships  
Relationships

New measure  
Calculations

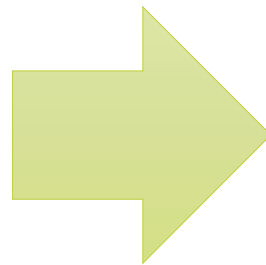
Quick measure  
Calculations

New column  
Calculations

New table  
Calculations

Structure

Date ID	Data Date
1	10 July 2023
2	18 March 2023
3	05 December 2023
4	17 May 2023
5	01 October 2023
6	03 October 2023
7	12 May 2023
8	30 March 2023
9	06 March 2023
10	01 June 2023
11	15 February 2023
12	26 June 2023
13	23 November 2023
14	19 July 2023
15	16 November 2023
16	11 August 2023
17	09 December 2023
18	23 December 2023
19	27 August 2023



File Home Help External tools **Table tools** **Column tools**

Paste Cut Copy

Get data

Excel workbook

OneLake data hub

SQL Server

Enter data

Data

Datawarehouse

Recent sources

Transform data

Refresh

Manage relationships

Clipboard

Queries

Relationships

Date ID	Data Date	Year	Month	Formatted Date	Quarter
1	10 July 2023	2023	7	2023-07-10	3
2	18 March 2023	2023	3	2023-03-18	1
3	05 December 2023	2023	12	2023-12-05	4
4	17 May 2023	2023	5	2023-05-17	2
5	01 October 2023	2023	10	2023-10-01	4
6	03 October 2023	2023	10	2023-10-03	4
7	12 May 2023	2023	5	2023-05-12	2
8	30 March 2023	2023	3	2023-03-30	1
9	06 March 2023	2023	3	2023-03-06	1
10	01 June 2023	2023	6	2023-06-01	2
11	15 February 2023	2023	2	2023-02-15	1
12	26 June 2023	2023	6	2023-06-26	2
13	23 November 2023	2023	11	2023-11-23	4
14	19 July 2023	2023	7	2023-07-19	3

# MEASURES

- Data panel
- Go to Fact table
- New Measure
- `msr_platinum_tier = CALCULATE(SUM(Fact_Sales[Total Spend]),FILTER(Dim_Customer,[Service Tier]="Platinum"))`
- Pag1 1
- Card

Sales ID	Date Time	Items	Total Spend	Product ID	City ID	Customer ID	Date ID
1	10/07/2023 18:37:38	19	2542	1	1	1	1
2	18/03/2023 23:28:40	1	4722	2	2	2	2
3	05/12/2023 22:16:30	1	9088	3	3	3	3
4	17/05/2023 20:59:08	3	9033	4	4	4	4
5	01/10/2023 12:58:00	12	5127	4	5	1	5
6	03/10/2023 15:58:24	1	7479	5	4	4	6
7	12/05/2023 15:53:59	3	1577	3	3	5	7
8	30/03/2023 20:18:39	27	8973	1	6	5	8
9	06/03/2023 19:42:29	4	7107	3	1	6	9
10	01/06/2023 03:07:43	2	1034	4	5	7	10
11	15/02/2023 03:49:29	24	3068	3	3	8	11
12	26/06/2023 12:49:19	8	5474	5	7	7	12
13	23/11/2023 18:17:21	6	2603	4	8	7	13
14	19/07/2023 07:56:30	23	5777	4	3	8	14
15	16/11/2023 10:14:13	22	2415	4	5	8	15
16	11/08/2023 09:20:58	21	379	5	2	9	16
17	09/12/2023 07:16:41	7	1020	5	2	6	17
18	06/03/2023 11:39:22	27	7845	1	9	3	18
19	23/12/2023 10:04:11	9	7040	3	2	10	19
20	27/08/2023 02:02:11	5	5931	3	1	1	19
21	21/08/2023 20:14:30	28	1126	3	6	3	20
22	31/01/2023 04:27:49	24	4195	5	8	2	21
23	28/11/2023 01:48:23	12	9838	2	6	7	22
24	24/03/2023 06:34:56	9	839	5	10	8	23
25	13/12/2023 05:54:24	19	6922	5	4	5	24
26	26/03/2023 19:38:35	8	5529	5	7	8	25
27	16/06/2023 23:56:17	26	1108	1	5	3	26
28	10/11/2023 15:06:16	27	4060	2	6	4	27
29	25/07/2023 09:35:46	19	1307	4	3	4	28
30	30/03/2023 11:09:04	7	262	5	1	3	8
31	19/10/2023 01:55:58	5	721	4	5	5	29

# MEASURES

- Data panel
- Go to Fact table
- New Measure
- `msr_customers_prod_items =  
CALCULATE(COUNT(Fact_Sales[Customer  
ID]),FILTER(Fact_Sales,(Fact_Sales[Product ID]=2 ||  
Fact_Sales[Product ID] = 1) && Fact_Sales[Items]>25))`
- Pag1 1
- Card > Fields > drag and drop the measure

# Copilot – Prompt

## Aggregation using DAX

Please generate the dax code that will allow me to group by Dim\_Customer[Customer] and Dim\_Product[Product] and then calculate the sum of Fact\_Sales[Total Spend] ,  
WHERE Dim\_City[Population Size] > 200000 and  
sort by customer and product

# Answer - Incorrect

```
EVALUATE
VAR FilteredSales =
    FILTER(
        ADDCOLUMNS(
            Fact_Sales,
            "Population Size", RELATED(Dim_City[Population Size])
        ),
        [Population Size] > 200000
    )
VAR GroupedSales =
    SUMMARIZE(
        FilteredSales,
        Dim_Customer[Customer],
        Dim_Product[Product],
        "Total Spend", SUM(Fact_Sales[Total Spend])
    )
RETURN
    SORTBY(GroupedSales, Dim_Customer[Customer], ASC, Dim_Product[Product], ASC)
```

# Answer – Correct

```
EVALUATE
VAR FilteredSales =
    FILTER(
        ADDCOLUMNS(
            Fact_Sales,
            "Population Size", RELATED(Dim_City[Population Size])
        ),
        [Population Size] > 200000
    )
VAR GroupedSales =
    SUMMARIZE(
        FilteredSales,
        Dim_Customer[Customer],
        Dim_Product[Product],
        "Total Spend", SUM(Fact_Sales[Total Spend])
    )
RETURN
    GroupedSales ORDER BY Dim_Customer[Customer] ASC, Dim_Product[Product] ASC
```

Results | Result 1 of 1 | Copy

	Dim_Customer[Customer]	Dim_Product[Product]	[Total Spend]
1	Customer A	Product A	2542
2	Customer A	Product B	10897
3	Customer A	Product D	5931
4	Customer B	Product A	14483
5	Customer B	Product D	1344
6	Customer B	Product E	6922
7	Customer C	Product A	1190
8	Customer C	Product B	18381



# Creating a table from DAX

## Don't include sorting

```
tbl_GroupByTest =  
VAR FilteredSales =  
    FILTER(  
        ADDCOLUMNS(  
            Fact_Sales,  
            "Population Size", RELATED(Dim_City[Population Size])  
        ),  
        [Population Size] > 200000  
    )  
VAR GroupedSales =  
    SUMMARIZE(  
        FilteredSales,  
        Dim_Customer[Customer],  
        Dim_Product[Product],  
        "Total Spend", SUM(Fact_Sales[Total Spend])  
    )  
RETURN  
    GroupedSales
```

The screenshot shows the 'Table tools' ribbon in Microsoft Power BI. The 'Name' field is set to 'tbl\_GroupByTest'. The 'Structure' pane on the left shows a table with three columns: 'Customer', 'Product', and 'Total Spend'. The table contains 10 rows of data.

Customer	Product	Total Spend
Customer A	Product A	2542
Customer J	Product C	4722
Customer J	Product B	9033
Customer H	Product E	12748
Customer E	Product D	16082
Customer C	Product E	5474
Customer C	Product B	18381
Customer I	Product E	379



# Summary

- Covered:
  - Data Modelling Concepts
    - Dimensional Modelling
  - Dimensional Modelling - Business Case
  - Designing a Data Model
    - Dimensional Model
  - Implementing a Data Model
    - Dimensional Model – Star Schema
  - Including Measures in the Data Model
  - Dax Aggregation within a Dimensional Model
  - Generating Reports