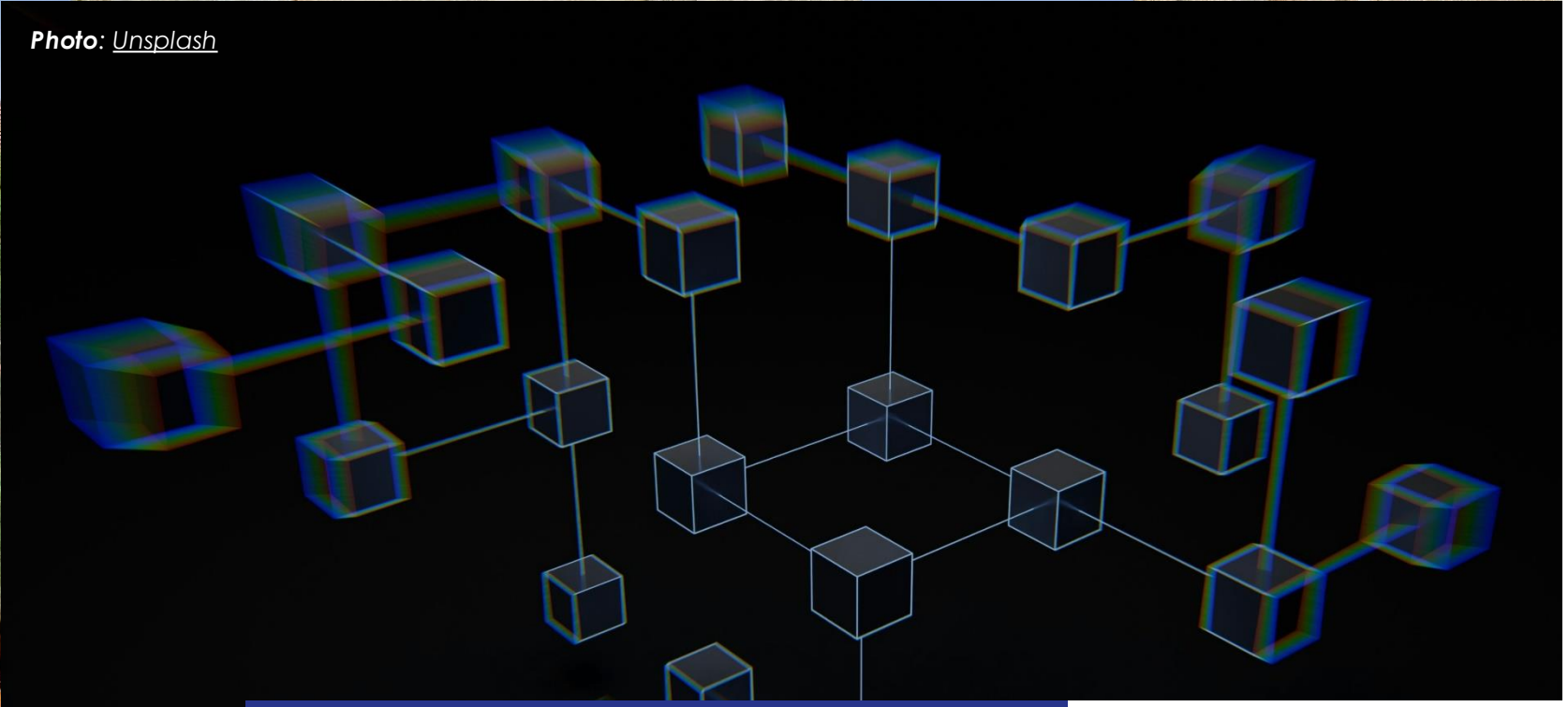


Photo: Unsplash

THE UNIVERSITY OF WESTERN AUSTRALIA

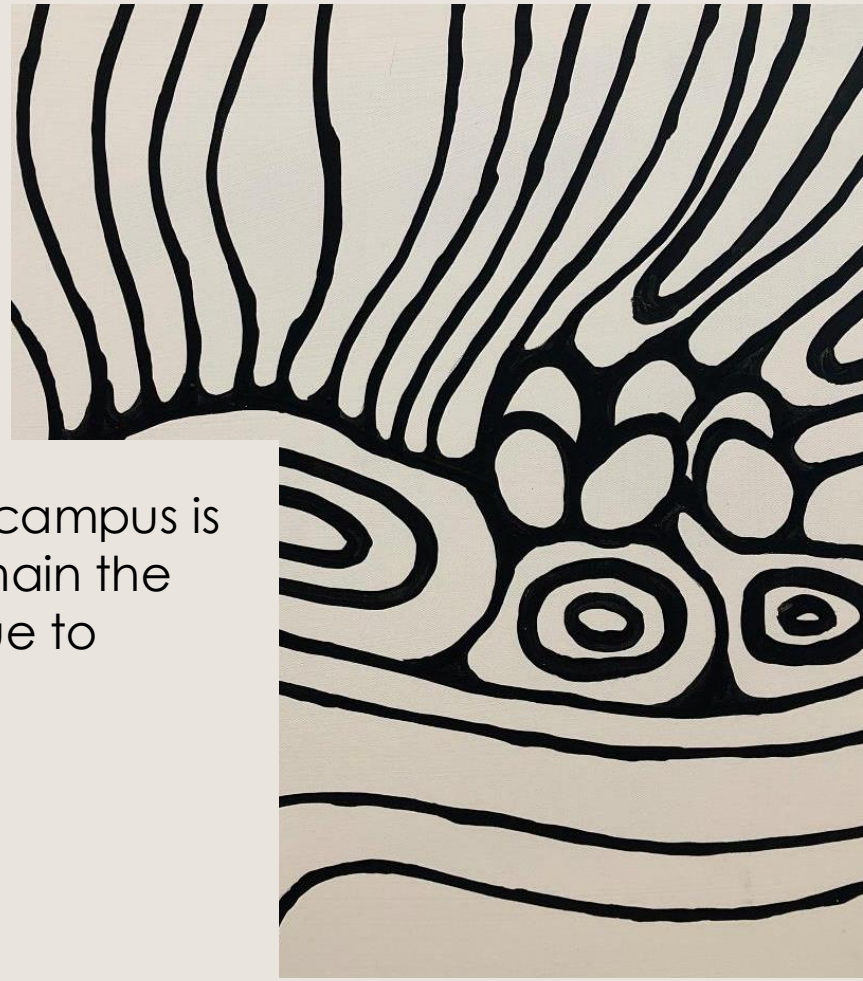
Topic Twelve: Revision

INMT5526: Business Intelligence

Tristan W. Reed
UWA Business School

Acknowledgement of country

The University of Western Australia acknowledges that its campus is situated on Noongar land, and that Noongar people remain the spiritual and cultural custodians of their land, and continue to practise their values, languages, beliefs and knowledge.



Artist: Dr Richard Barry Walley OAM

What is Business Intelligence?

- Tableau (2022), a provider of business intelligence analysis and visualisation software tools suggests that it "*combines business analytics, data mining, data visualization, data tools and infrastructures and best practices to help organizations make more data-driven decisions*".
- Everyone seems to have a different definition – over time, the generally accepted definition has changed from "data sharing" to similar to this, based upon the idea of "modelling for making decisions".

Tableau. (2022). *Business Intelligence: A complete overview | Tableau*. [online]. Available from: https://www.tableau.com/learn/articles/business-intelligence.

Descriptive, Predictive and Prescriptive

- Analytics are often termed in a "three" level hierarchy (insightsoftware, 2022):
  - Descriptive: describing what happened (in the past);
  - *Diagnostic: helps explains why something happened (in the past);*
  - Predictive: predicting what is likely (could) happen (in the future);
  - Prescriptive: suggests how can we make these outcomes happen (in the future);
  - *Comparative*: comparing two different groups in various ways.

Insightsoftware. (2022). *Comparing Descriptive, Predictive, Prescriptive and Diagnostic Analytics - insightsoftware*. [online]. Available from: https://insightsoftware.com/blog/comparing-descriptive-predictive-prescriptive-and-diagnostic-analytics/.

Interrogating Data

- Often we will want to look at different components of our data or look at it in different ways.
  - Drilling up and drilling down refer to the ability to explore the next level of detail in a report or visual – but done first in a Database Management System.
  - Drill through refers to the ability to select a part of a visual and be taken to another page in the report, filtered to the data that relates to the part selected on the original page.
  - A filter removes data that does not apply whereas a highlight greys out the data that does not apply.

Databases vs Spreadsheets

THE UNIVERSITY OF WESTERN AUSTRALIA

- The main difference between databases and spreadsheets are that spreadsheets are designed for simple data manipulation tasks undertaken by a single user (at least a single user at a time).
  - Databases can handle multiple people interrogating the data at the same time;
  - Databases provide greater performance for larger data sets;
  - Databases also provide finer-grained access controls.
- Can you think of a benefit a spreadsheet has over a database?

Data Model Formats in a Database

THE UNIVERSITY OF WESTERN AUSTRALIA

- Traditional (and most common) *relational* databases model data in terms of *tables* of *rows* and *columns*.
  - Other data models exists that don't do this – e.g. NoSQL document databases;
  - A database can contain many tables of 'infinite' rows and columns, but there are practical and logical considerations to this.
  - This format is generally used due to the efficiency and regularity of the format – it is known each row has the same columns and vice versa.
  - We can manipulate these with SQL and a database management system (DBMS).

Business Decision Making & Databases

THE UNIVERSITY OF WESTERN AUSTRALIA

- Together with BI tools (which can be linked to databases), the power of today's computer systems can allow organisations to analyse this data.
  - This can be used to help businesses run more efficiently, make better decisions and react to changes quicker (agility) and grow (scale) quicker.
  - Decisions can only be make quickly "in time" when the storage, access and analysis of the data is easy, consistent and fast - otherwise by the time the analysis is completed, the insights or decisions can be out of date. Hence, the DBMS used must be performant.
  - Volumes of data just continue to grow and grow – where do we go next? Web 3.0?

Database Keys

THE UNIVERSITY OF WESTERN AUSTRALIA

- "Keys" are used in relational databases as an attribute to uniquely identify individual observations (rows) from each other.
  - As such, the values must be unique – otherwise more than one row can be identified!
  - Primary keys (PK)'s are stored within a table, whereas a Foreign key (FK) is a reference to a PK in another table other than the current table.
  - This therefore creates a relationship between the two tables such that the second table can be looked up from the first and considered to 'extend' the values of (and the) attributes defined within it.

A Relationship is Born

THE UNIVERSITY OF WESTERN AUSTRALIA

- If the value of a foreign key in a row of one table defines a row in another table, a relationship between two tables is created.
  - How we term this relationship depends on how it is formed.
- The various *cardinalities* of relationships (and their abbreviations) are below:
  - *One to Many (1:M)*: One record in a table is linked to many records in another table.
  - *Many to One (M:1)*: Many records in other tables are linked to one record in a table.
  - *One to One (1:1)*: *One* record in one table is linked to exactly one record in another.

Creating a Database

THE UNIVERSITY OF WESTERN AUSTRALIA

- Using MySQL Workbench, run the following command to create a database named "**<StudentID>_db**":
  - **CREATE DATABASE <StudentID>_db;**
- Of course, you must substitute in <u>your</u> student ID where **<StudentID>** is listed in the above query!
  - Only for our own server within this unit!

MySQL Data Types

THE UNIVERSITY OF WESTERN AUSTRALIA

- The (major) data types and their uses are as follows:
  - **`VARCHAR(X)`** : a *variable length* string of X characters (benefits?);
  - **`CHAR(X)`** : a fixed length string of X characters;
  - **`BOOLEAN`**: a **`TRUE`** or **`FALSE`** value (stored as **`0`** or **`1`**);
  - **`INT`**: an integer (whole) number;
  - **`FLOAT`**: a floating point (with decimal portion) number;

Attribute (Field) Modifiers

THE UNIVERSITY OF WESTERN AUSTRALIA

- There are a few modifiers (there are many more) that we can add:
  - **`PRIMARY KEY`** to specify that a field(s) is a primary key;
  - **`AUTO_INCREMENT`** to automatically increase an integer field for each row;
  - **`NOT NULL`** to ensure a value is specified for each observation (row);
  - **`FOREIGN KEY`** to specify a foreign key relationship (next week's job).
- MySQL goes about this a bit unusually in case of the keys.
  - These are specified at the end of the statement, rather than in the 'middle' like they are with other DBMS – one of the quirks that I was talking about!

Modifying a Table

THE UNIVERSITY OF WESTERN AUSTRALIA

- We may want to adjust our data model after we have created it.
  - However, we must consider any issues this creates with respect to the integrity of the data within the table – especially if we add a new column.
- This is achieved through an **`ALTER TABLE`** command:
  - **`ALTER TABLE <TableName> ADD COLUMN ColumnName varchar(255);`** to add a new column to **`<TableName>`**;
  - **`ALTER TABLE <TableName> DROP COLUMN <ColumnName>;`** to remove a column from **`<TableName>`**;

Modifying (Altering a Field)

THE UNIVERSITY OF WESTERN AUSTRALIA

- Alternatively (pun somewhat intended) we can change or modify one of the columns without recreating it:
  - **`ALTER TABLE <TableName> CHANGE <OldColumnName> <NewColumnName> <Modifiers>;`** if we wish to change the name and certainly the modifiers of the column as well;
  - Just the name: **`ALTER TABLE <TableName> RENAME <OldColumnName> TO <NewColumnName>;`**
  - Just the modifiers: **`ALTER TABLE <TableName> MODIFY <ColumnName> <Modifiers>;`**

Deleting a Table

THE UNIVERSITY OF WESTERN AUSTRALIA

- We may wish to get rid of a table – but we must be careful!
  - This will delete all of the data within the table, not just the model itself!
- We achieve this through a **`DROP TABLE`** command.
  - **`DROP TABLE <TableName>;`**
- This is why we should keep track of the commands that we issue to manipulate our database!
  - You may wish to do so in a Word document (or similar)!

Changing/Updating Data

THE UNIVERSITY OF WESTERN AUSTRALIA

- The command to update (change) data in MySQL is relatively straightforward and similar to the command to *create* data in MySQL.
  - **`UPDATE <TableName> SET <ColumnNameOne> = <NewValueOne>, <ColumnNameTwo> = <NewValueTwo>`**… `WHERE <ColumnName> = <Value>;`
- We can specify a **`WHERE`** clause after this (before the semicolon as shown in navy) to only affect certain rows only.
  - We would want to do this for at least e.g. **`id = 1`** so we only affect one row.

Reading Data

THE UNIVERSITY OF WESTERN AUSTRALIA

- To read (or view) data within the table in MySQL, we use the "Select" statement in one of the two ways seen below:
  - **`SELECT <ColumnNameOne>, <ColumnNameTwo> FROM <TableName>;`**
  - **`SELECT * FROM <TableName>;`**
- We can also view a subset of our data by specifying a **`WHERE`** clause in the same manner as with changing data.

Deleting Data

THE UNIVERSITY OF WESTERN AUSTRALIA

- The command to delete data is relatively similar to deleting a table.
  - **`DELETE FROM <TableName> WHERE <ColumnNameOne> = <ValueOne>…;`**
- If we do not specify a **`WHERE`** clause, then everything in the table is deleted;
  - Tread carefully – you may wish to do a **`SELECT`** first to test!

Adding Data

THE UNIVERSITY OF WESTERN AUSTRALIA

- The command to insert (add/create) data within a table is relatively straightforwards:
  - **`INSERT INTO TableName (ColumnNameOne, ColumnNameTwo…) VALUES (ValueOne, ValueTwo...);`**
- This would create a single new record within TableName where the value of **`ColumnNameOne`** is **`ValueOne`** and the value of **`ColumnNameTwo`** is **`ValueTwo`**.
  - Repeat for however many columns that we have – only required if **`NOT NULL`**.

Viewing (Reading) a Table

THE UNIVERSITY OF WESTERN AUSTRALIA

- We can examine the data types and attribute (field/column) names of our table by issuing the following command:
  - **`SHOW COLUMNS FROM <TableName>;`**
- Remember that we must specify the database name as part of the "table name" (e.g. **`12345678_db.<TableName>`**)!

Generated Columns: Process

THE UNIVERSITY OF WESTERN AUSTRALIA

- The main keyword 'phrase' used to effect the creation of a generated column is **`GENERATED ALWAYS AS (<function>);`** which is used as a modifier on a column.
  - Where **`<function>`** involves one or more other columns.
  - We can add **`VIRTUAL`** or **`STORED`** before the semicolon to decide when to calculate it.
  - The **`CONCAT()`** function can be used to combine one or more text fields (columns/attributes), however mathematical operators can also be used on numeric types.
- Column names only need to be specified on their own.
  - The table and/or database names are not needed!

Additional Modifiers / Constraints

THE UNIVERSITY OF WESTERN AUSTRALIA

- There are also some other modifiers to constrain values added to particular attributes, which we can do (apply) to various differing fields.
  - **`DEFAULT <defaultValue>`** to specify a default value of **`<defaultValue>`** for the field.
  - **`CHECK (<formula>)`** as an additional item in the attribute list to ensure that a constraint described by **`<formula>`** must be satisfied before a value is added (or changed!) for a particular column(s).
- We can name a constraint in the format **`CONSTRAINT <name> CHECK (<formula>)`** if we wish to do so for clarity or simplicity.

Complex 'WHERE' Operators

THE UNIVERSITY OF WESTERN AUSTRALIA

- We can use the following operators (ignoring the colons) to compare values of a field (attribute) in a more complex way than just 'is equal to'.
  - `=`: is equal (the same) – won't work for floating point values;
  - `>`: greater than (numeric types) or `<`: less than;
  - `>=`: greater than or equal to or `<=`: less than or equal to;
  - `<>`: Not equal to (sometimes written as `!=`);
  - `BETWEEN`: between a certain range (specified – `x AND y`);
  - `LIKE`: basic pattern matching (using `%` as a wildcard);
  - `IN`: matches a value from a comma-separated list of values.

Complex ‘Where’ keywords

THE UNIVERSITY OF WESTERN AUSTRALIA

- To make things even more finer-grained (albeit complicated), we can combine formula by using the following commands:
  - **`AND`**: both the left and right side must be true to match;
  - **`OR`**: either (or both of) the left and right side must be true to match;
  - **`NOT`**: only applied to one formula, gets the opposite result (what is true is false).
- **`NOT`** can be combined with the other two.
  - However, in either case – use plenty of brackets for clarity!

Foreign Key Relationships

THE UNIVERSITY OF WESTERN AUSTRALIA

- Much in the manner we define primary keys, we can also define foreign keys.
  - The 'other' table and its primary key must first already exist before a foreign key relationship can be made to it from a different table.
- We add in the format of an 'extra' attribute **`FOREIGN KEY (<AttributeName>) REFERENCES <TableName>(<ColumnName>)`**.
  - We can also name it through instead describing this as **`CONSTRAINT <ConstraintName> FOREIGN KEY <AttributeName> REFERENCES <TableName>(<ColumnName>)`**.
  - This allows us to define multiple-attributed foreign keys as well!

Comment Formats for Queries

THE UNIVERSITY OF WESTERN AUSTRALIA

- The three types of comments are as follows:
  - Starting with a `#` (hash symbol) - anything else until the next line becomes a comment.
  - Starting with a `--` (two dashes) - anything else until the next line becomes a comment.
  - Between a `/*` and `*/` character sequences – anything between these two (which can be over multiple lines) becomes a comment.
- A reminder: comments are not executed as queries!
- You are free to choose which format is appropriate and preferred.
  - It is suggested that you comment your work through the exercise, where relevant.

Show Create Table Query / Command

THE UNIVERSITY OF WESTERN AUSTRALIA

- We recall that the **`CREATE TABLE`** command is used to create our data model.
  - We specify the data type, modifiers and names of each of our attributes, alongside the name of our table (and the database it resides in).
  - We can use an **`ALTER TABLE`** command to add/change/remove these attributes.
- Through the use of a **`SHOW COLUMNS FROM`** command we can see some of the above information – but not all of them.
  - This makes it hard to see what we have already done to our table!

Sorting Syntax

THE UNIVERSITY OF WESTERN AUSTRALIA

- To sort our data, we add a sorting clause on the end of our **`SELECT`** query, before the semicolon but after a **`WHERE`** clause (if we have it).
- After the words **`ORDER BY`**, we list (comma-separated) the attributes (columns) we wish to sort by, first to last.
- We can then state **`ASC`**(ending) or **`DESC`**(ending) depending on which direction that we'd like to sort by (for each attribute).
- For example: **`SELECT * FROM SomeTable WHERE id > 0 ORDER BY SomeColumn ASC;`**

How We Group

THE UNIVERSITY OF WESTERN AUSTRALIA

- The **`GROUP BY`** clause sits between the **`WHERE`** clause and the **`ORDER BY`** clause in our **`SELECT`** query.
  - One or more columns are then specified which are the ones that will form the grouping.
  - The values for the first column are grouped, then the second and so forth.
  - The same caveats apply as they did with the sorting (**`ORDER BY`**) clause – consider a relevant number of groups that are meaningful to sort the problem.
  - You can see how this would be most useful for text, then date, then whole (integer) numeric and not very useful for floating point numeric.

The Actual Aggregating Functions

THE UNIVERSITY OF WESTERN AUSTRALIA

- The five most common aggregating functions that will be used are:
  - **`AVG(<ColumnName>)`** to get the mean of the values in **`<ColumnName>`**;
  - **`MIN(<ColumnName>)`** to get the smallest of the values in **`<ColumnName>`**;
  - **`MAX(<ColumnName>)`** to get the largest of the values in **`<ColumnName>`**;
  - **`SUM(<ColumnName>)`** to get the total of the values in **`<ColumnName>`**;
  - **`COUNT(<ColumnName>)`** to get the count of the values in **`<ColumnName>`**;
- It can be seen that these would only be appropriate for particular data types.

Date Functions

THE UNIVERSITY OF WESTERN AUSTRALIA

- Within our **`GROUP BY`** clause, we may only want to group on part of a date:
  - **`MONTH(<ColumnName>)`** to group by the month ONLY;
  - **`YEAR(<ColumnName>)`** to group by the year ONLY;
- We also have **`NOW()`** as a shortcut (useful for **`WHERE`**).
- There exists many more date arithmetic functions.
  - You can read about them at https://dev.mysql.com/doc/refman/8.0/en/date-and-time-functions.html.

Types of Joins

THE UNIVERSITY OF WESTERN AUSTRALIA

- There are four ways we can join a table, depending on which rows from which tables we wish to keep (as it won't be guaranteed there are always matches):
  - **`LEFT JOIN`**: all observations from the first table and matching from the second;
  - **`RIGHT JOIN`**: matching observations from the first table and all from the second;
  - **`INNER JOIN`**: matching observations that are within both tables;
- The most appropriate type of join depends on the context of the problem being solved.

Joins – Syntax Format

THE UNIVERSITY OF WESTERN AUSTRALIA

- The format for joining two tables (lets call them **TableA** and **TableB**) within MySQL is relatively straightforward:
  - ```
    SELECT TableA.AttributeA, TableB.TheID, TableB.AttributeA FROM TableA
    INNER JOIN TableB ON TableA.TheID = TableB.TheID;
    ```
- We can see it is the presence of the **INNER JOIN** keyword that differs.
  - **<SecondTable> ON (TableB)** syntax specifies the 'other' table we wish to join;
  - The equation below is very simple - 'this attribute in this table equals that';
  - Note the prefix of the tables and which columns have been selected.
    ```

Creating Point Data

THE UNIVERSITY OF WESTERN AUSTRALIA

- In MySQL, we simply provide the spatial data type when we define an attribute in a **`CREATE TABLE`** or **`ALTER TABLE`** statement, i.e.:
  - `CREATE TABLE GeometryTable (geometry POINT);`
- We can then insert the spatial data itself using the following format as our VALUE (there are other ways as well):
  - `ST_GeomFromText('POINT(<Long> <Lat>)')`
- We must ensure that the data that is inserted is *valid* – what makes it so?

Spatial Queries

THE UNIVERSITY OF WESTERN AUSTRALIA

- Due to the nature of spatial data, we are able to write more complex queries that take advantage of this information.
  - For example, finding locations within X kilometres from a specified location.
- A very simple (is it?) query we can write is to filter for locations within 10 kilometres of a given point:
  - `SELECT * FROM <TableName> WHERE ST_Distance_Sphere(<GeomField>, ST_GeomFromText(<Location>)) <= 10 * 1000 ORDER BY name;`

Creating Data in MongoDb

THE UNIVERSITY OF WESTERN AUSTRALIA

- We must first create a collection ("table") to store our many documents:
  - Command is **`db.createCollection("<CollectionName>")`** for **`<CollectionName>`**;
- We can then add one or more documents to the collection:
  - Takes the format **`db.<CollectionName>.insert({attribute: value, attribute: value, …})`**, where **`<CollectionName>`** is the collection you created.
  - You can specify a list of dictionaries as well – i.e. **`insert([{…}, {…}])`** to save time.
  - It is preferred to use **`insertOne`** and **`insertMany`** for these.

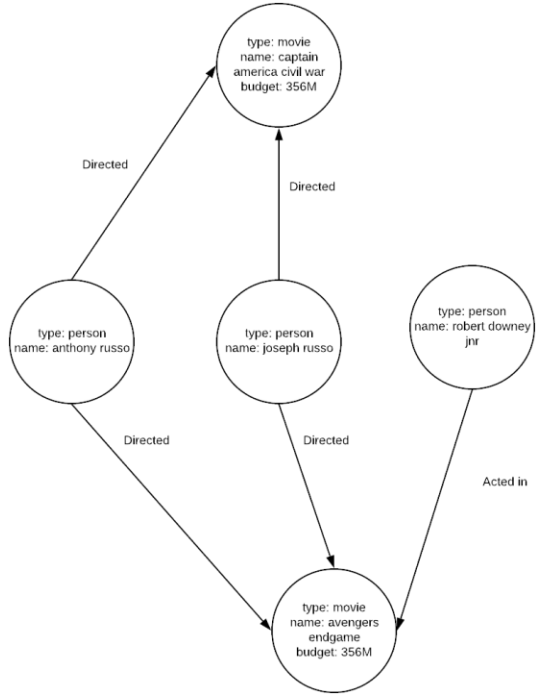
Inequalities in MongoDb QUeries

THE UNIVERSITY OF WESTERN AUSTRALIA

- We use various **`$keywords`** to use inequalities in our queries:
  - **`.find({attributeName: {$lt: 1}})`**: less than one;
  - **`.find({attributeName: {$lte: 1}})`**: less than or equal to one;
  - **`.find({attributeName: {$gt: 1}})`**: greater than one;
  - **`.find({attributeName: {$gte: 1}})`**: greater than or equal to one;
  - **`.find({attributeName: {$ne: 1}})`**: not equal to one;

Graph databases

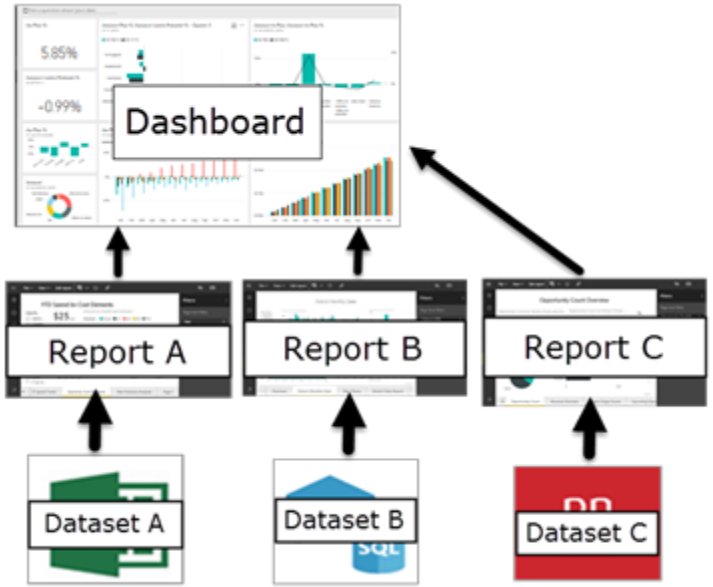
- A graph database is based on the mathematical concept of a graph.
  - A collection of nodes (entities – observations) and edges (directed connections with a meaning that relate two nodes together).
  - Technically, this means both nodes and edges hold data!



Clarke, J. (2019). *An Introduction to Graph Databases | by John Clarke | Introduction to Data Science*. [online] Available from: https://towardsdatascience.com/an-introduction-to-graph-databases-cd81a0d5aa12.

Reports and Dashboards

- *Recall* that a report is a view on a single dataset.
- A dashboard can be composed of multiple views.
- A collection of reports and/or dashboards is called an App.
- **Can you think of some issues with calling this package an App?**



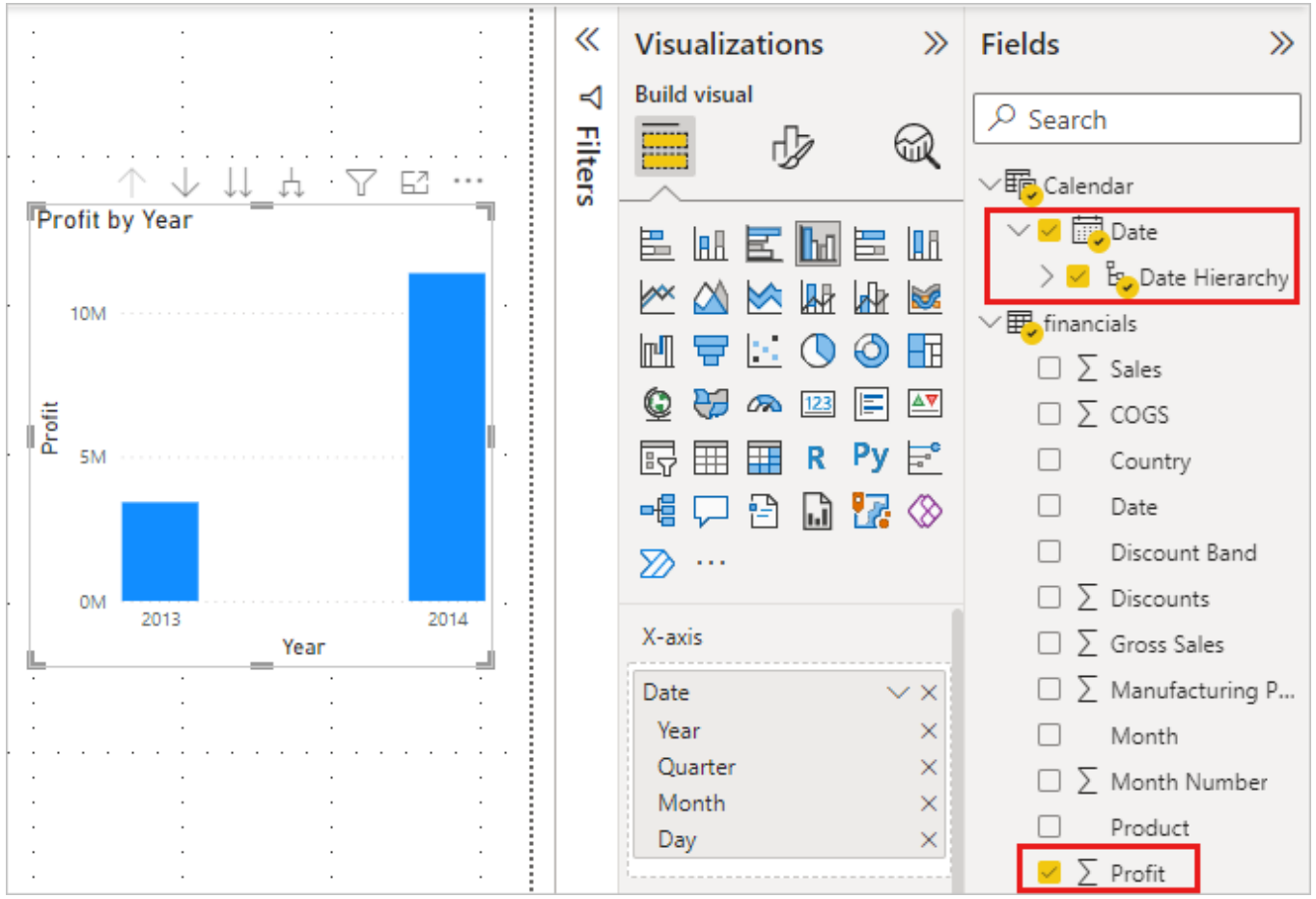
Microsoft. (2022). *Introduction to dashboards for Power BI designers– Power BI | Microsoft Learn*. [online]. Available from: https://learn.microsoft.com/en-us/power-bi/create-reports/service-dashboards.

Transforming Data

THE UNIVERSITY OF WESTERN AUSTRALIA

- Often, we will need to *transform* our data before we can use it. This can take many forms of operations we can undertake:
  - Deriving one attribute from one or more attributes;
  - Adjusting the shape of the data ("rows and columns");
  - Adjusting the data type/data format of an element.
- We do this to ensure that the data is regular, in the most efficient and useful format and to ensure that the data follows the 'usual' format – rows & cols.

Creating the Visuals



We can select a Visualisation from the Visualisations pane to add it to the Report.

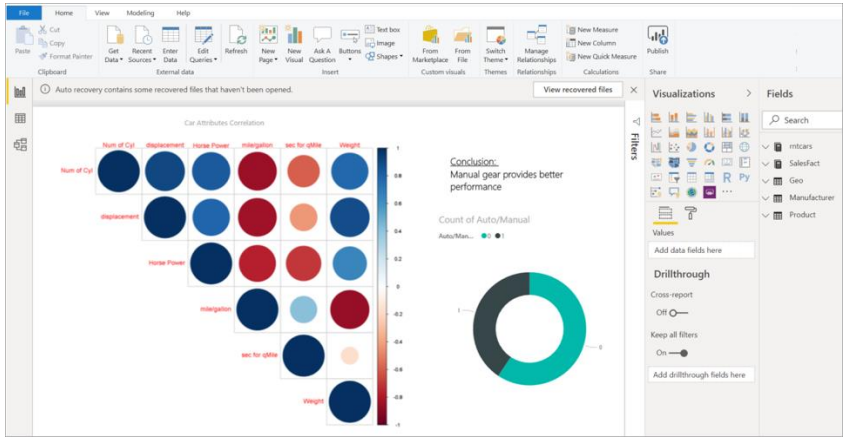
Selecting the Visualisation within the report, the fields can then be selected as seen.

Additional options at the bottom.

Creating a Correlation Plot

THE UNIVERSITY OF WESTERN AUSTRALIA

- If we immediately add the following code below to generate a correlation matrix and then create a correlation plot of it, we will run into a problem.
- Add the following two lines below to do so:
  - ```
    corr_matrix <- cor(dataset)
    corrplot(corr_matrix)
    ```


    ```

Imported vs Connected

- Imported data provides us a ‘snapshot in time’ with a normal file (residing on our system) that is only changed by us.
  - Useful if we want to consider a snapshot in time.
- Connected data comes from a remote system and can be updated as new events occur and the data source is updated.
  - Connected data allows us to refresh the dashboard and update it with the latest data.
  - Enables us to see how/if our conclusions change.

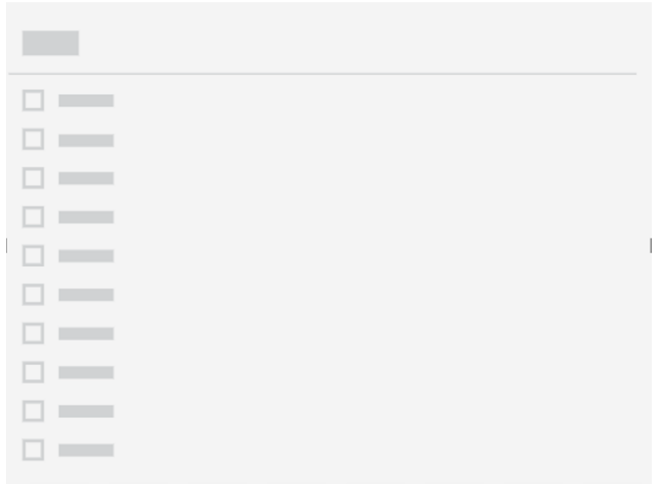
Business Intelligence Workflow

- Everyone will have a different opinion on the specifics of this style of 'data analysis workflow', but the following provides a good summary:

- It is often said that the majority of time is spent in the first three steps of this process!

Drilling up and down

- Not all are strictly visualisations; consider our slicer to filter / highlight / drill data.
- We must ensure that our data is mutually exclusive for our slicer to function in a practical way.
- We looked at other ‘not quite’ visualisations last week!



Security Issues

THE UNIVERSITY OF WESTERN AUSTRALIA

- Unprotected systems can leak data – that is, provide the data to those that don't need (and should not have) access to it!
  - Ensure that you are using a strong username and password;
  - Minimal permissions to each group that has them – internet availability;
  - Keep software up to date to ensure bugs/exploits are patched;
  - Ensure audit logs and technical solutions are enabled and utilised.

Ethical Issues

- Data can be used for less than proper purposes, not those which the consumer has consented to – is this right to do?
  - Targeted advertising based upon previous behaviour – limited surveillance;
  - Prediction of health outcomes (e.g. insurance) based upon some form of personal data;
  - Automatic prediction of outcomes from data – 'stereotypes';
  - Using data for purposes not originally consented to;
  - Email mailing list sign ups – laws should protect against this.

Privacy Issues

- Private data should remain just that – it is unethical to provide this data to those who don't need it and for purposes other than agreed to.
  - Private data required for legitimate business purposes;
  - Erosion of this privacy may cause fraud or embarrassment;
  - Technical controls should be used to achieve this – alongside social ones!

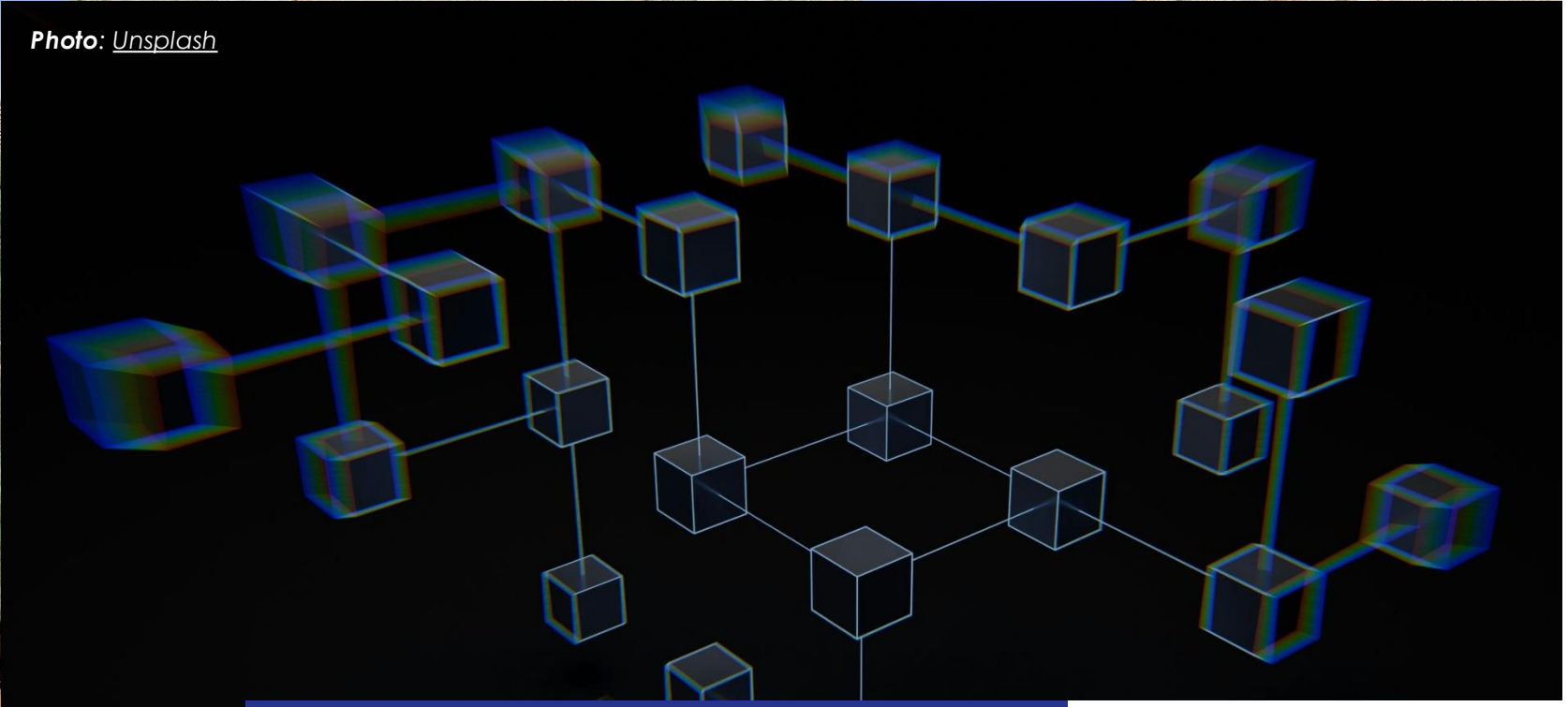


Photo: Unsplash

Topic Twelve: Future of Databases

INMT5526: Business Intelligence

Tristan W. Reed
UWA Business School

Blockchain

THE UNIVERSITY OF WESTERN AUSTRALIA

- A distributed, immutable ledger of transactions.
  - **Distributed**: the source of truth is distributed between multiple nodes (machines);
  - **Immutable**: past transactions can't be changed;
  - **Ledger of Transactions**: list of standardised operations undertaken.
- We have seen many use cases for Blockchain.
  - Generally in the fields of Cryptocurrency, NFT's, Smart Contracts…
  - Good use cases solve problems that other system's cant – emphasis on above properties!

That's All, Folks!

THE UNIVERSITY OF WESTERN AUSTRALIA

- This is it for the semester, at least for the lecture part of things.
  - Don't forget this weeks' lab, with similar discussions to what we did before.
  - There's also a couple of assessments still outstanding – including SparkPlus.
- Thank you, goodbye and best of luck for your future endeavours.
  - It has been a pleasure to teach you all and to learn about you and your journey.
  - Don't forget to complete the student feedback survey (if you so desire – it's optional).
  - As I always like to say, "until we meet again" – which seems to happen a lot!
- Don't hesitate to email me if you ever have a quick query!

The End: Thank You

Any Questions? Ask via email (tristan.reed@uwa.edu.au)