Further Practice – Section 1

1. Create a new, blank database and save it in the same folder that you are working in as **Elements**. Close the database.

2. Open the database **Chemistry** and open the **Elements** table. One of the columns is not wide enough to display the full heading. Widen the appropriate column.

3. Define the **Atomic Number** field as the **Primary Key** for the table.
   Change the **Field Size** for the following fields:

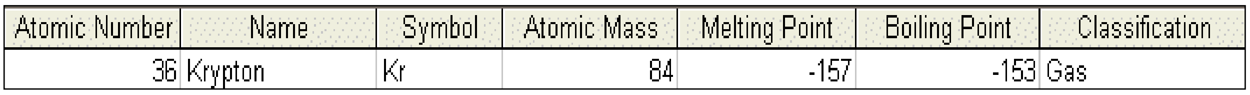
   Name: 20 characters
   Symbol: **2** characters

4. Add a validation rule to the **Classification** field so that it will only accept entries of **Metal**, **Solid**, **Liquid** or **Gas**.
   Set the **Validation Text** to: **Please enter Metal, Solid, Liquid or Gas**

5. Add a validation rule to the **Atomic Number** field so that values greater than **100** cannot be entered.

6. Filter the table to show only **Metal** elements. Sort the filtered data in ascending order of **Melting Point**. Print a copy of the filtered table in portrait orientation then remove all filters/sorts. (Tip: Use the Remove Sort command in the **Home** tab/**Sort and Filter** group).

7. Add a new record to the **Elements** table:



Atomic Number	Name	Symbol	Atomic Mass	Melting Point	Boiling Point	Classification
36	Krypton	Kr	84	-157	-153	Gas

8. Create a query from the **Elements** table to show **all** fields from those elements classified as **gases**, sorted in **alphabetical** order of element **Name**. Save the query as **Gases**.

9. Create another query from the **Elements** table to show the fields in the following order:
   Name **Atomic Number** **Atomic Mass** **Symbol**
   Display all elements <u>**not**</u> classified as **Gases**. Sort the query results in descending order of **Atomic Mass** and save as **Not Gases**.

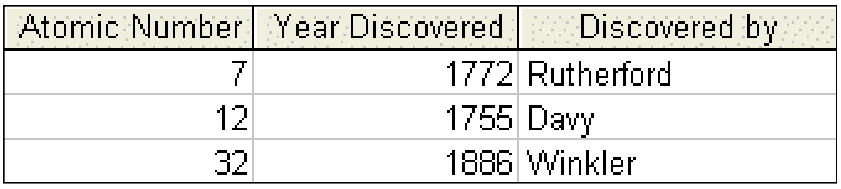
10. Print a copy of the **Gases** query and the **Not Gases** query in **Landscape**.

11. Create a new query to show **Atomic Number**, **Name**, **Symbol** and **Melting Point** for all elements which melt between **0** and **100** degrees Celsius. Save the query as **Melt**.

12. Create a new table with three fields:

Field Name	**Data Type**	**Format or Field Size**
Atomic Number	Number	Long Integer
Year Discovered	Number	Long Integer
Discovered by	Text	40

13. Save the table as **History** and make **Atomic Number** the primary key. Add the following three records to the **History** table:



Atomic Number	Year Discovered	Discovered by
7	1772	Rutherford
12	1755	Davy
32	1886	Winkler

14. Create a relationship between the **Elements** table and the **History** table based on the **Atomic Number** field. What type of link is created? Enforce **referential integrity**.

15. Use the wizard to create a **columnar form** including all the fields from the **Elements** table. Save the form as **Element Data**.

16. Go to the design view of the form. Add a title of **Element Data Form** to the **Form Header** area. Include your name at the left edge of the **Form Footer** and save the form.

17. Use the **Element Data** form to find the record for **Sulphur**. Ensure the paper size is **A4** and print a copy of the form for the **Sulphur** record only.

18. Use the wizard to create a **landscape stepped report** showing all the fields from the **Elements** table, grouped by **Classification**. Ensure all data is fully displayed and save the report as **List**. Close the **Chemistry** database.

19. You've just been informed that a database you have previously created already exists. Delete the **Elements** database.

Further Practice – Section 2

1. Which of the following is *not* a common use of a large scale database?
    a) maintenance of hospital patient details
    b) calculation of office accounts
    c) maintenance of government records
    d) airline booking systems

2. Open the database **Hire**. Some items on the database are obsolete. Delete the table **Cars**, the form **Car Details** and the report **Stock**.

3. Open the **Vehicles** table in datasheet view, showing details of some of the available vehicles owned by a small car hire company. Make sure all data and headings are displayed in full.

4. Add a validation rule to the **Type** field so that it will only accept entries of **Compact**, **Family** or **Sports**. Set the **Validation Text** to list the allowed values in the event of an invalid entry being made.

5. Add a new field to the end of the **Vehicles** table called **Charge**. The data type is to be **Currency** with **0** decimal places. Apply a validation rule to the **Charge** field so that no charge greater than **150** is allowed.

6. Add **Charge** data to all existing records on the table according to the following table: (Tip: You might find it easier to sort the table by type before you add this information and then remove the Sort option once you have entered the records)

Type	**Charge**
Compact	50
Family	75
Sports	90

7. Create a new table called **Bookings** with no primary key, to hold details of vehicle bookings as follows:
   (Tip: if you receive an alert message for the first field, ignore this)

Field Name	Data Type	Format or Field Size
Name	Text	30 characters
Vehicle Number	Number	Long Integer
Start Date	Date	Short Date
Number of Days	Number	Integer
Deposit Paid	Yes/No	Default value **Yes**

8. Create a relationship between the new **Bookings** table and the **Vehicles** table based on the only field that is common to both tables. Apply referential integrity to the relationship.

9. Use the wizard to create a columnar **form** including all the fields from the **Bookings** table. Save the form as **Booking Form**.

10. Edit the **Booking Form** so that a title of **Booking Entry** appears in the **Form Header** area. Save the form and use it to enter the following bookings for vehicles 211, 212, 214 and 314.

Name	Vehicle Number	Start Date	Number of Days	Deposit Paid
G Khan	211	29/05/21	3	Yes
A Smith	212	29/05/21	4	No
D McKenna	214	02/06/21	5	No
P Tell	211	20/05/21	3	Yes
J Kirk	314	03/06/21	7	Yes

11. You will notice that the last record (J Kirk) will not enter on the database. This is because there is no record for Vehicle number 314 in the Vehicles table! The referential integrity prevents this record from being entered. The vehicle number **314** is incorrect, it should be **313**. Re-enter this record and replace **314** with **313**.

12. Create and run a query which lists all **Compact** vehicles from the **Vehicles** table. Show all fields from the table. Save the query as **Compact**. Print a copy of the result of the **Compact** query. (Note: keep a screen capture of this query as evidence as you will be later asked to delete this query.)

13. Create a query with fields **Vehicle Number**, **Start Date** and **Number of Days** from the **Bookings** table, together with **Type** and **Charge** from the **Vehicles** table. Save the query as **Enquiry**. Print a copy of the **Enquiry** query in **Landscape** orientation.

14. Delete the **Compact** query.

15. Create a **Landscape** orientation report based on all the fields from the **Vehicles** table grouped by **Type** of vehicle and sorted by **Vehicle Number**. Save the report as **Types** and print a copy. Edit the **Types** report. Change the report header to **Vehicle Listing** and remove the **Seats** field (and its heading) from the report.

16. Create a **Landscape** orientation report based on all the fields from the **Enquiry** query. Using the default grouping options, group the report by **Start Date**. Show the sum of **Number of Days** as a summary total. Save the report as **Booking List**.

17. Edit the report so that in the **Page Footer** area, the date appears on the left and your name appears on the right. Remove any other content from the area.

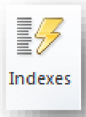
18. Save the report and print a copy then close the **Hire** database.

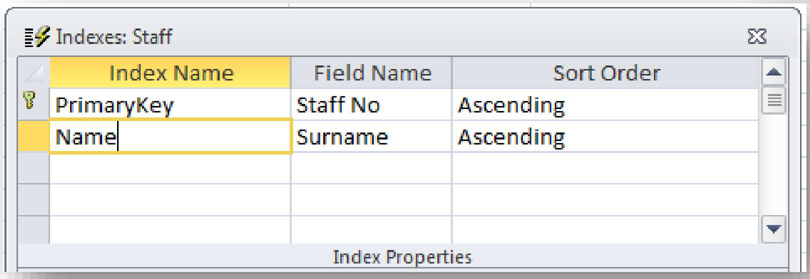
Further Practice – Section 3

1. Open the database **Wages** and open the **Staff** table. In what sequence are the **Staff** records shown by default?

2. Use **Help** to find out the function of indexes.

   Go to the design view and create an index called **Name** based on the **Surname** field and define ascending order. (**Design** Tab/**Show/Hide** group)

   Indexes: Staff

Index Name	Field Name	Sort Order
PrimaryKey	Staff No	Ascending
Name	Surname	Ascending

   Close the **Indexes:Staff** window.

   In the **Surname** field, set the Index properties to **duplicates not allowed** (note this will not automatically reorder the data). Do <u>not</u> replace the **Primary Key** index. Save the table.

3. Filter the table to show only employees in the **Production** department.

4. Sort the filtered table in ascending order of **Data of birth**. Remove the filter.

5. Filter the table to show only employees who are <u>not</u> in the **Production** department.

   Sort the filtered table in descending order of **Rate**. Remove all filters/sorts.

6. Create a query showing all fields from the **Staff** table, for all employees who have started the company since **1st January 2020**. Save the query as **New Starters**.

7. Create a query to see if there are any employees in the **Production** department who joined the company before the year **2000**. Show all fields from the **Staff** table in the resulting list. Save the query as **20 Years service**.

8. Create the following table to hold details of the hours worked by each employee.

Field Name	**Data Type**	**Format**
Staff No	Number	Long Integer
Date	Date	Short Date
Hours Worked	Number	Integer

Save the table as **Hours** with **no** primary key.

9. Create a relationship between the **Hours** table and the **Staff** table based on an appropriate field from each table.

10. Create a **landscape** report, based on the **Staff** table, which is grouped by **Department**, and sorted on **Staff No**. Save the report as **Staff List**.

11. The **Staff List** report will need to be edited so that the title **Department Listing** is shown in the report header.

12. Paul Branson has left the company. Locate his record in the table and delete it. Add your own details with **Staff No 21**. Assign yourself to the **Production** department with an hourly rate of **£20**.

    How would you print this record only?

13. Print a copy of the **Staff List** report with your name included and then close the **Wages** database.