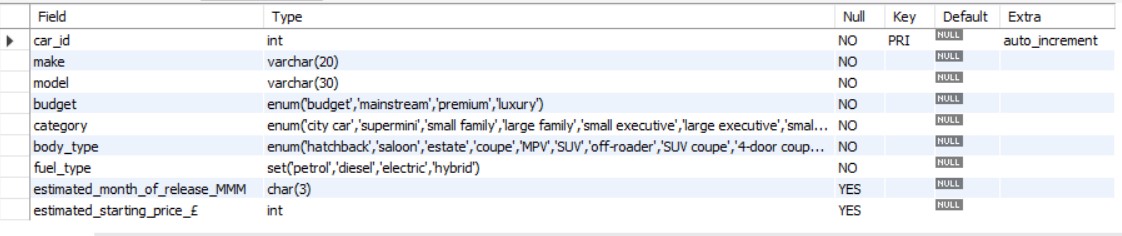
SQL Task One:

The subject chosen for the database was ‘new car releases for 2022’. The initial fieldnames chosen for table one (new car releases) were car ID, make, model, category, body type, fuel type and estimated month of release, with ‘car ID’ being the primary key and modified to automatically increment. After this, the fieldnames ‘estimated starting price’ and ‘budget’ were added. The ‘budget’ fieldname came after ‘estimated starting price’ so, using the ‘alter table’ command, ‘budget’ was specified to come after ‘model’. All fieldnames apart from ‘estimated month of release’ and ‘estimated starting price’ used the ‘not null’ modifier to ensure that values were input. The remaining two fieldnames allowed null values as it is not necessarily specified which month they will be released or how much they will cost. The table structure can be seen below:



The reasons for the field type for each field are as follows:

Car ID – only numbers would be input

Make – the names of each make are not a fixed length

Model – the names of each model are not a fixed length

Budget – each record could have only one possible answer from a given list

Category - each record could have only one possible answer from a given list

Body type - each record could have only one possible answer from a given list

Fuel type – each record could have multiple answers from a given list

Estimated month of release – each record must have a three-letter abbreviation for the month

Estimated starting price – only numbers would be input.

Following on from this, the data was input into the table. This was done in separate parts because some records did not have a release month and/or a starting price.

Graphical user interface, text, application

Description automatically generated

The ‘SELECT \* FROM’ command was used throughout to ensure there were no errors and again at the end to view the completed table. In the screenshot below can be seen examples where the fieldnames ‘estimated month of release’ and ‘estimated starting price’ have a ‘null’ value:

Graphical user interface, application

Description automatically generated

SQL Task Two:

For table two (Telford Motor Show), the subject chosen was cars being exhibited at a motor show. The initial fieldnames chosen for the table were car ID; make; model; concept or production; location; stand; available for test drive; and type of test drive, with ‘car ID’ being the primary key and modified to automatically increment. After this, the fieldnames ‘fuel type and ‘test drive stand’ were added with ‘fuel type’ being moved to come after ‘concept or production’ using the ‘alter table’ command. All fieldnames apart from ‘type of test drive’ and ‘test drive stand’ used the ‘not null’ modifier to ensure that values were input. The remaining two fieldnames allowed null values as there would be no values if the cars were not available for test drive. To avoid ambiguity, ‘location’ and ‘stand’ were renamed ‘locations’ and ‘show stand’ (stand could be confused with test drive stand and for many records there were more than one location). The table structure can be seen below:

Graphical user interface, text, application, email

Description automatically generated

The reasons for the field type for each field are as follows:

Car ID – only numbers would be input

Make – the names of each make are not a fixed length

Model – the names of each model are not a fixed length

Concept or production – each record could have only one possible answer from a given list

Fuel type - each record could have multiple answers from a given list

Locations - each record could have multiple answers from a given list

Show stand – each record could have only one possible answer from a given list

Available for test drive – each record could have only one possible answer from a given list

Type of test drive– each record could have multiple answers from a given list

Test drive stand– each record could have multiple answers from a given list.

Before inputting any data, the fieldname ‘model’ for both tables were modified to only allow unique entries. This was due to a car having different car IDs in each table (all data was input alphabetically in each instance). This would then allow for queries once the tables were joined. The updated table structures can be seen below:

Table One structure

Graphical user interface, text, application, email

Description automatically generated

Table Two structure

Graphical user interface, text, application, email

Description automatically generated

Following on from this, the data was input into the table. To make inputting easier, records that had no data for specific fieldnames were input using ‘null’, as can be seen below:

A picture containing chart

Description automatically generated

The ‘SELECT \* FROM’ command was again used throughout to ensure there were no errors and again at the end to view the completed table. Due to inputting data using a different method, the fieldnames with null values were left blank. In the screenshot below can be seen examples of this:

Table

Description automatically generated

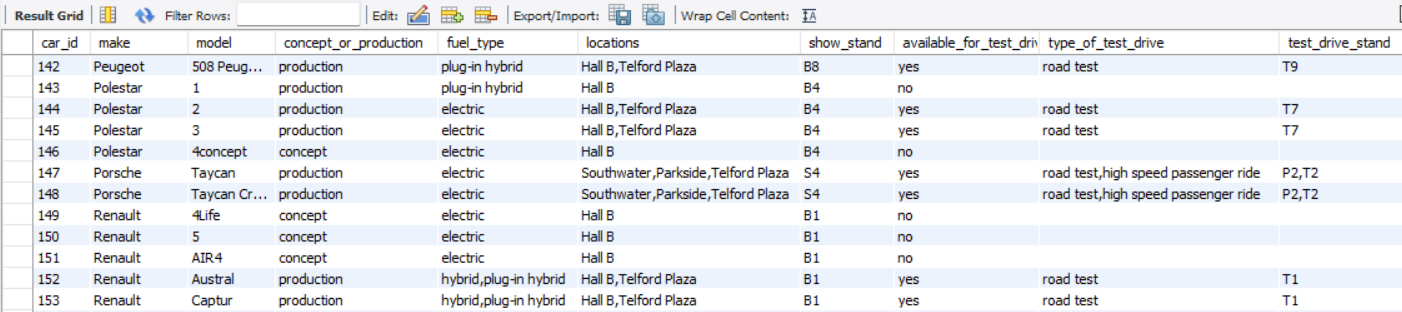
After analysing the data, errors were identified. The affected records were then updated using the UPDATE, SET, WHERE commands. The two screenshots below show the original and updated data for records 147 and 148 (locations; type of test drive; and test drive stand):

Original data

Graphical user interface, text, application, email

Description automatically generated

Updated data



Upon further research, the release date for one of the cars in table one had been moved to some time after 2022 which meant that the record (42) needed to be deleted. This was done using the ‘DELETE FROM’ command. The screenshots below show the table before and after this was completed:

Before deleting record

Graphical user interface, text, application, email

Description automatically generated

After deleting record

Text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Once both tables were complete, a selection of simple queries on table two were carried out as can be seen in the following screenshots:

Concept cars

Graphical user interface, table

Description automatically generated

Cars with an off-road experience

Graphical user interface, table

Description automatically generated

The tables were then joined together:

Graphical user interface, text, application, email

Description automatically generated

The following complex queries were then carried out:

New cars releases for 2022 available at the Telford Motor Show for test drive

Table

Description automatically generated

Electric cars at the Telford Motor Show that are new for 2022

Table

Description automatically generated

To show which new car releases were not at the motor show, the ‘SELECT, FROM, WHERE, NOT IN’ command was completed:

Graphical user interface, application, table

Description automatically generated

Two errors were identified: car IDs 3 and 9. Car ID 3 had been omitted and car ID 9 had been typed incorrectly. The errors were fixed using the ‘add’ and ‘modify’ functions before using the ‘SELECT, FROM, WHERE, NOT IN’ command again:

A picture containing text

Description automatically generated

Graphical user interface, application

Description automatically generated

The ‘SELECT, FROM, WHERE, NOT IN’ command was used again to show which companies with new releases were absent from the motor show:

Graphical user interface, text, application, email

Description automatically generated

Each table was then sorted by various parameters, the most relevant for each can be seen below:

Table One results ordered by price

Graphical user interface, application, table

Description automatically generated

Table Two results ordered by show stand

Graphical user interface

Description automatically generated

Using comparison operators, data was filtered as follows:

New car releases with a price between (and including) £20,000 and £40,000

Graphical user interface, text, application, email

Description automatically generated

Cars located on stands in the Exhibition Centre (Halls A, B and C)

Graphical user interface

Description automatically generated with low confidence

\*To do this, stands between (and including) A1 and C6 were used in the search.