**Assignment 1 - Question 2A**

**Conceptual Design:**

**Seller review Entity:**

|  |  |  |
| --- | --- | --- |
| **Seller review number** | **Reviews about the sellers**  **( \* Poor, \*\* Fair, \*\*\* Good,**  **\*\*\*\* Very Good,**  **\*\*\*\*\* Excellent)** | **Date time feedback submitted** |
|  |  |  |
|  |  |  |
|  |  |  |

**Product review Entity:**

|  |  |  |
| --- | --- | --- |
| **Product review number** | **Reviews about products**  **(\* Poor, \*\* Fair, \*\*\* Good,**  **\*\*\*\* Very Good,**  **\*\*\*\*\* Excellent)** | **Date time feedback submitted** |
|  |  |  |
|  |  |  |
|  |  |  |

**Question Entity:**

|  |  |
| --- | --- |
| **Question about products** | **Date time question asked** |
|  |  |
|  |  |
|  |  |

**Answer Entity:**

|  |  |
| --- | --- |
| **Answers for the questions** | **Date time answers posted** |
|  |  |
|  |  |
|  |  |

Here are the three entities that will be used in my table design to store the additional information needed from the requirements.

**High-level entity relationship diagram (ERD):**

****

This is a basic high level entity relationship diagram that I will use for my table design.

**Logical Design:**

Here I assign a primary key to each of my entities:

**Seller review entity:**

|  |  |  |
| --- | --- | --- |
| **Seller review number** | **Reviews about sellers**  **(\* Poor, \*\* Fair, \*\*\*Good,**  **\*\*\*\* Very Good,**  **\*\*\*\*\* Excellent)** | **Date time feedback submitted** |
|  |  |  |
|  |  |  |
|  |  |  |

In this entity I didn’t have anything that was unique to the Seller reviews entity, so I added a new INTEGER column and made that the primary key (Seller review number).

**Product review entity:**

|  |  |  |
| --- | --- | --- |
| **Product review number** | **Reviews about products**  **(\* Poor, \*\* Fair, \*\*\*Good,**  **\*\*\*\* Very Good,**  **\*\*\*\*\* Excellent)** | **Date time feedback submitted** |
|  |  |  |
|  |  |  |
|  |  |  |

Like with the previous entity I did not have anything that was unique to the product reviews entity, so I added a new INTEGER column and made that the primary key (Product review number).

**Question entity:**

|  |  |  |
| --- | --- | --- |
| **Question number** | **Question about products** | **Date time question asked** |
|  |  |  |
|  |  |  |
|  |  |  |

I also did the same thing here that I did for the previous two entities as, again, there were no pre-existing columns in this entity that were fit to be the primary key, so I created one (Question number).

**Answer entity:**

|  |  |  |
| --- | --- | --- |
| **Answer number** | **Answers for questions** | **Date time answers posted** |
|  |  |  |
|  |  |  |
|  |  |  |

And finally, I did the same thing here that I did for the previous entities as there were no pre-existing columns in this entity that were fit to be the primary key, so I made one (Answer number).

**Final Logical Design:**

Here is a more in depth ERD for the logical design of my tables, with the primary keys underlined and foreign key marked with an \*.

****

**Physical Design:**

**Seller review table:**

|  |  |  |
| --- | --- | --- |
| **Review number** | **Reviews\_about**  **\_sellers**  **(\* Poor, \*\* Fair, \*\*\*Good,**  **\*\*\*\* Very Good,**  **\*\*\*\*\* Excellent)** | **Date\_time\_feedback**  **\_submitted** |
|  |  |  |
|  |  |  |
|  |  |  |

I will add a CHECK constraint on the Date\_time\_feedback\_submited column to ensure that it only accepts dates in this format: DD/MM/YYYY, and time in this format: HH/MM/SS.

As well as this I will add a CHECK constraint on the Reviews\_about\_sellers column to ensure that only VARCHAR data types can be inputted into them and no other data type (i.e, INTEGER, FLOAT, BOOLEAN etc.)

Also I will add a CHECK constraint on the Date\_time\_feedback\_submitted column to ensure that only DATETIME data types can be inputted into it and no other data type (i.e, VARCHAR, FLOAT, BOOLEAN, INTEGER etc.)

I will also use another CHECK constraints on the Reviews\_about\_sellers column to ensure that no more than 5 \*’s are inputted and at least 1 \* is inputted to ensure that anyone who reviews a seller does so in the correct way.

The primary key will increase automatically by one every time another piece of data is added to the table, and it will never change as to keep it unique.

**Product review table:**

|  |  |  |
| --- | --- | --- |
| **Review number** | **Reviews\_about\_products**  **( \* Poor, \*\* Fair, \*\*\* Good,**  **\*\*\*\* Very Good,**  **\*\*\*\*\* Excellent)** | **Date\_time\_feedback**  **\_submitted** |
|  |  |  |
|  |  |  |
|  |  |  |

I will add a CHECK constraint on the Date\_time\_feedback\_submited column to ensure that it only accepts dates in this format: DD/MM/YYYY, and time in this format: HH/MM/SS.

As well as this I will add a CHECK constraint on the Reviews\_about\_products column to ensure that only VARCHAR data types can be inputted into them and no other data type (i.e, INTEGER, FLOAT, BOOLEAN etc.)

Also I will add a CHECK constraint on the Date\_time\_feedback\_submitted column to ensure that only DATETIME data types can be inputted into it and no other data type (i.e, VARCHAR, FLOAT, BOOLEAN, INTEGER etc.)

I will add another CHECK constraint to make sure that if the Reviews\_about\_products column has data in it, then so must the Date\_time\_feedback\_submitted column.

I will also use another CHECK constraints on the Reviews\_about\_products column to ensure that no more than 5 \*’s are inputted and at least 1 \* is inputted to ensure that anyone who reviews a product does so in the correct way.

Like the seller reviews entity the primary key will automatically increase by one every time another piece of data is added to the table, and it will never change as to keep it unique.

**Question table:**

|  |  |  |
| --- | --- | --- |
| **Question\_number** | **Questions\_about\_products** | **Date\_time\_questions\_asked** |
|  |  |  |
|  |  |  |
|  |  |  |

I will add a CHECK constraint on the Questions\_about\_products column to ensure that only VARCHAR can be inputted into it and no other data type (i.e, INTEGER, FLOAT, BOOLEAN etc.)

As well as a CHECK constraint on the Date\_time\_questions\_asked to ensure that only DATETIME data types can be inputted into it and no other data type (i.e, TEXT, FLOAT, BOOLEAN etc.)

I will also add a CHECK constraint on the Date\_time\_questions\_asked column to ensure that it only accepts dates in this format: DD/MM/YYYY, and time in this format:

HH/MM/SS.

I will also add a CHECK constraint to make sure that if the Questions\_about\_products column has data in it, then so must the Date\_time\_questions\_asked column.

Like with the previous tables, the primary key will also increase automatically by one every time another piece of data is added to the table and it will never change as to keep it unique.

**Answer table:**

|  |  |  |
| --- | --- | --- |
| **Answer\_number** | **Answers\_for\_questions** | **Date\_time\_answers\_posted** |
|  |  |  |
|  |  |  |
|  |  |  |

I will add a CHECK constraint on the Date\_time\_answers\_ posted column to ensure that it only accepts dates in this format: DD/MM/YYYY, and time in this format: HH/MM/SS

I will also add a CHECK constraint on the Date\_time\_answers\_ posted to ensure that only INTEGER data types can be inputted into it and no other data type (i.e, TEXT, FLOAT, BOOLEAN etc.)

As well as a CHECK constraint on the Answers\_for\_questions column to ensure that only TEXT can be inputted into it and no other data type (i.e, INTEGER, FLOAT, BOOLEAN etc.)

I will also add a CHECK constraint to make sure that if the Answers\_for\_questions column has data in it, then so must the Date\_time\_answers\_posted column to ensure that every answer has a date and time it was posted.

Like with the other tables the primary key will also increase automatically by one every time another piece of data is added to the table, and it will never change as to keep it unique.

**Physical Design:**

****

Here the entity titles have been pluralised to create the table name, Review to Reviews, Answer to Answers and Question to Questions. And any spaces in the column names are replaced by an underscore.

**Final Physical Design:**

This is the final Physical Design of my table with the data types included in it as well as foreign (indicated with \*) and primary keys (primary keys are underlined).