Introduction to the system

Online Market's database design is used to manage website's data. In this database system users are customers. They can buy products online, place their purchase in a shopping cart, choose their payment type, and leave reviews. This database contains all information about its products and suppliers.

In this Database Management System project, customers can buy products online and the administrator can enter the name and generate the receipt of the purchased product. Administrators can view reports of the products and overall order details. Orders which are placed by the customers, will store into the database and according to the order detail, a bill will be generated and the payment will be paid by the customer.

ER diagram

Entities:

user_info
user_address
address
country
payment_type
payment_type_method
shopping_basket
shopping_basket_item
product_item
product
product_category
variation
variation_option
product_var
shop_order
order_details
order_status
order_line
User_comment

Coding part

We created a procedure that groups the data in the order and outputs the quantity and total amount of the product.

And a procedure that counts rows whose prices are increased to 20$.

Function that counts the number of items in the basket which are chosen more than one time.

Exception that disallows entering the name of a product to be less than* 5 characters.

Trigger that shows number of users before inserting a new row.

Normal form

In 1NF, a relation (or table) is said to be in first normal form if it meets the following requirements:

1. No repeating groups: Each column in a table should contain only one value for each row. There should be no repeating groups of columns.
2. Unique column names: Each column in a table should have a unique name.
3. Unique rows: Each row in a table should be unique. There should be no duplicate rows.

A table is said to be in second normal form if it meets the following requirements:
1. It is in first normal form.
2. All non-key columns in the table are fully dependent on the entire primary key.

A table is said to be in third normal form if it meets the following requirements:
1. It is in second normal form.
2. There are no transitive dependencies.

The **"USER_INFO"** table

1.1NF

The USER_INFO table satisfies the requirements for 1NF because each column contains atomic values and there are no repeating columns.

2.2NF

Since this table only has a single primary key (USERID) it is automatically in 2NF. There are no partial dependencies in this table.

3.3NF

There are no transitive dependencies in this table because each non-key column (FIRSTN, LASTN, PHONEN, EMAILN, and PASSWORDN) is directly dependent on the primary key (USERID). Therefore, the table satisfies the requirements for 3NF.

1NF.The **"ADDRESS"** table may contain the addresses of users, and in this case, several data may be recorded in one row. To avoid this problem, we have a table "user address", this table helps us exactly to which address of the user we need to send orders and also only one given one is recorded in each row.

2NF. Each column in a table have unique name.

3NF. There are not duplicate rows.

The **"COUNTRY"** table

1NF

The COUNTRY table satisfies the requirements for 1NF because each column contains atomic values and there are no repeating columns.

2NF

Since this table only has a single primary key (COUNTRYID) it is automatically in 2NF. There are no partial dependencies in this table.

3NF

There are no transitive dependencies in this table because each non-key column (COUNTRYN) is directly dependent on the primary key (COUNTRYID). Therefore, the table satisfies the requirements for 3NF.

The **"PAYMENT_TYPE"** table
1NF: The table has a primary key column (PAYMENT_TYPE_ID) with unique values, and each column contains atomic values.

2NF: The table has only one candidate key, which is the primary key, and there are no partial dependencies.

3NF: The table has no transitive dependencies, and all non-key attributes depend only on the primary key.

The "**PAYMENT_TYPEMETHOD**" table

1NF: Each column contains atomic values, there are no repeating groups, and each row is uniquely identifiable by a primary key.
2NF: The table is in 1NF.This table only has a single column primary key, so there are no partial dependencies.
3NF: The table is in 2NF, and there are no transitive dependencies where a non-key attribute is dependent on another non-key attribute. This table also meets this requirement since PROVIDER, CARD_NUMBER, and CVV are all directly dependent on the primary key, and USERID and PAYMENT_TYPE_ID are foreign keys.

<u>In many tables, primary keys identify the non-prime attributes along with the foreign keys, but this does not mean that there are transitive dependencies in these tables</u>

The "**PRODUCT_CATEGORY**" table
1NF: All columns in the table hold atomic values, and there are no repeating columns.

2NF: The table is in 1NF and all non-key attributes in the table are fully dependent on the primary key. This table has only one primary key, CATEGORYID, and there is only one non-key attribute, CATEGORYN, which is fully dependent on the primary key.

3NF: The table is in 2NF and all non-key columns in the table are independent of each other. Since there is only one non-key attribute in this table, it is automatically independent of any other non-key attributes that might exist in the table. Therefore, this table is also in 3NF.

The “**PRODUCTT”** table

1NF.Every column of the table contains atomic values, meaning that the values cannot be further divided. The primary key is also defined, and there are no repeating groups or arrays of data stored in the table.

2NF. The PRODUCTT table has a single-column primary key (PRODUCTID), which means it meets the requirements for 2NF.

The CATEGORYID column is a foreign key that references the CATEGORYID primary key column in the PRODUCT_CATEGORY table, which ensures that there is no partial dependency in the PRODUCTT table.

All non-key columns (CATEGORYID, PRODUCTName, and PRODUCT_DEScription) are fully dependent on the primary key (PRODUCTID). This means that each non-key column is dependent on the entire primary key, rather than just part of it, so there is no partial dependency.

PRODUCTID ->PRODUCTName,PRODUCT_DEScription

3NF. The table appears to be in 3NF because there are no transitive dependencies between non-key columns.PRODUCTID (Primary Key): This column uniquely identifies each row in the table. PRODUCTName: It is directly dependent on the primary key (PRODUCTID) and not on any other column in the PRODUCTT table.

PRODUCT_DEScription: It is directly dependent on the primary key (PRODUCTID) and not on any other column in the PRODUCTT table.

PRODUCTID->PRODUCTNAME, PRODUCTNAME->PRODUCTDES ==
~~PRODUCTID->PRODUCTDES~~

The CATEGORYID column is a foreign key that references the CATEGORYID primary key column in the PRODUCT_CATEGORY table, so there is no transitive dependency between CATEGORYID and any other non-key column.

The “**VARIATION**” table

1NF: All the columns are atomic and there are no repeating columns.

2NF: There is only one candidate key, VARIATIONID, which is a primary key. Also, there is a full functional dependency between the non-prime attribute, VARIATION, and the primary key, VARIATIONID, via the primary key, so it meets the requirements of 2NF.

Category_id foreign key for variation table.

It can not identify any value in variation table itself, because it is repeated

3NF: There are no transitive dependencies. The only non-prime attribute, VARIATIONN, depends directly on the primary key, VARIATIONID, via a foreign key, CATEGORYID, which references another table.

The "**VARIATION_OPTION**" table

1NF: All columns contain atomic values, and each row is unique.

2NF: The table is already in 1NF and every non-prime attribute is fully functionally dependent on the primary key. Here, the primary key is VARIATION_OPTIONID, and OPTIONV is fully dependent on it. VARIATIONID is dependent on the primary key

VARIATIONID foreign key for **VARIATION_OPTION**table.

It can not identify any value in **VARIATION_OPTION**table itself, because it is repeated

3NF: A table in 2NF, and no non-prime attribute is transitively dependent on the primary key. Here, there is no transitive dependency between the columns.

The "**PRODUCTT_ITEM**" table

1NF: Each attribute contains only atomic values, and there are no repeatingcolumns

2NF: The table is in 1NF ,the PRODUCTID and ITEMID together form the primary key and all other columns in the table are dependent on the primary key.

3NF: The table is in 2NF and, all the columns in the table are directly dependent on the primary key and there is no transitive dependency.

The table "**PRODUCT_VAR"**

1NF.It has only two columns, and both of them are single-valued attributes, so it meets the requirements of 1NF. There are no composite or multi-valued attributes.

2NF.The table has a composite primary key consisting of two foreign key columns, which are both not nullable, so it meets the requirements of 2NF.

3NF.Since there are no non-key attributes and the primary key consists only of foreign keys, the table is also automatically in 3NF.

The "**SHOPPING_BASKET"**

1.in the first normal form (1NF) as each column contains only atomic values, and there are no repeating columns.

2,3.There are no non-key attributes in this table, so it automatically satisfies the second normal form (2NF) and the third normal form (3NF).

The “**SHOPPING_BASKET_ITEM”** table

1NF: All the columns in the SHOPPING_BASKET_ITEM table contain atomic values, and there are no repeating columns

2NF: The table has a composite primary key consisting of BASKET_ITEMID, which is unique for each row, and SHOPPING_BASKETID, which is a foreign key that references the primary key of the SHOPPING_BASKET table. Additionally, ITEMID is a foreign key that references the primary key of the PRODUCTT_ITEM table. All non-key attributes are dependent on the entire composite primary key.

3NF: There is no transitive dependency between non-key attributes. All non-key attributes are directly dependent on the composite primary key.

The “**SHOP_ORDER**” table

1NF. All the columns in the **SHOP_ORDER**table contain atomic values, and there are no repeating columns

2NF. All non-key attributes depend on the key attribute. Here the key attribute is "ORDERID".

Users can repeat themselves in this table, many users can make an order in one day

3NF. There is no transitive dependency between non-key attributes. All non-key attributes are directly dependent on the composite primary key.

The “**ORDER_LINE**” table

1NF. All the columns in the **ORDER_LINE**contain atomic values, and there are no repeating columns

2NF. All non-key attributes depend on the key attribute. Here the key attribute is "ORDERED_ID".

It's the same here as in the table “**SHOP_ORDER**”,that is, data that is taken from other tables can be repeated

3NF. There is no transitive dependency between non-key attributes. All non-key attributes are directly dependent on the composite primary key.

The “**USER_COMMENT**” table

1NF. All the columns in the **USER_COMMENT**atomic values, and there are no repeating columns

2NF. All non-key attributes depend on the key attribute. Here the key attribute is "COMMENT_ID".

Data that is taken from other tables can be repeated, one user can write many comments

3NF. There is no transitive dependency between non-key attributes. All non-key attributes are directly dependent on the composite primary key.

The "**ORDER_STATUS**" table

This table is in 1NF, 2NF, and 3NF. It has a single primary key (STATUSID), and each column contains only atomic values.

There are no repeating groups or nested structures in this table, and no transitive dependencies.