Database Design & Manipulation (Includes SQL)

This project aims to get the proper skills to design and manipulate the Database using normalisation, transformation, SQL.

1. Project Context

The context of this project is the same as for first project(1.DatabaseModelling&Design), namely *The Magic Ale* (MA). This has been reproduced as is in the Appendix for convenience.

We pretend that your firm had an agreement with MA to design and implement the Database, and the job fell on your lap. You in turn assigned that task to an intern you are mentoring. The intern has gone through the specification and designed some relational schemas for this purpose. You had a quick look at them, and one of them particularly stood out since it had quite a number of attributes that you suspected should best belong to different relations. The suspect relation schema in question is:

```
Abnormal_Rel( ProductID, BranchID, campaignID, MemberID,
ProductType, PackageType, YearProduced, Price, Brand,
StockLevel, CampaignStartDate, CampaignEndDate, FirstName,
LastName, eMail, MembershipLevel, MemberExpDate, Discount

             )
```

Upon discussion with the intern you gathered that different branches can have the same product (`ProductID)` in different quantities. Also, in different campaigns, the same product may be given different discount rates for the same membership level.

You took upon yourself to explain to the intern the issue at hand, and how the issue should be addressed. Complete the following tasks in that context.

2 . Task Specifications

Task 1

Identify the non-trivial FDs on the relation Abnormal_Rel. Then identify the Candidate key(s) of Abnormal_Rel.

Task 2

Determine for each update anomaly whether or not the relation Abnormal_Rel is susceptible to that anomaly. Support your determination with adequate explanation and a small example.

Task 3

Determine the highest normal form that the relation Abnormal_Rel is in. Then:

1. Normalize/decompose it until you get relations that are in 3NF. Use appropriate illustration to aid the understanding of your work.
2. Check if the resultant relations are in BCNF. If not, decompose them as necessary until you get all of them in BCNF.

Task 4

Now (at the end of completing Task 3) you have a set of relation(s) in BCNF, derived from the relation Abnormal_Rel.

1. Create an appropriate table for each of these relations (in BCNF), keeping the key constraints in mind. Copy and paste into your .doc document the SQL code you used for this purpose.
2. Insert five rows of (made-up) data into each table. **Make sure** that the data you enter in these tables should be sufficient to return at least one row for each query in Task 5. For instance, MA should hold at least 5 bottles of Penfold Grange 2010 in some branch or other. Copy and paste into your .doc document the SQL code you used for this purpose.
3. Display the content of each table using a SELECT * query. Copy and paste into your .doc document the result that was displayed.

Task 5

Write SQL queries to answer the following four questions (Q1-Q4). With respect to each,

• Copy and paste into your .doc document the SQL query you used for this purpose.

• Display the result of running this query. Copy and paste into your .doc document the result that was displayed.

[Query 1] List the branches (ID) of MA that have in stock at least 5 bottles of Penfold Grange 2010.

[Query 2] Simone Singh plans to do some last-minute Christmas shopping on 24/12/2021. List details of each beer that she will be entitled to get 20% discount on.
[Query 3] Generate a list of all email addresses of members whose card will expire in the month after the coming month. Thus, for instance, if the query is run in November 2121, it will list the emails of all members whose membership will expire in January 2122. The emails should be ordered by Branch ID, then by expiry date, and then by the email address, all in ascending order.

[Query 4] Determine how many times Penfold Grange 2010 has gone on sale since Covid-19 related lockdown started (assume it to be March 01, 2020).

Appendix: Problem Context from first project.

This project concerns a liquor shop chain in Sydney, called *The Magic Ale* (MA). The objective of this project is to develop a database system that will be used to centrally store and manage all relevant information for the branches of MA.

The information to be stored include information on different branches of MA (Bankstown, Hornsby, etc.), types of drinks they sell (beers, wines, cedars, etc.), staff they employ (Retail Assistants, Shelving Assistants, etc.), Magic Members (MA Loyalty Card holders), and Sales Campaigns (discounts on specific products over a limited period). The basic requirements gathered from the stake holders is presented in the following five points. As typically the case, these requirements are often underspecified. Use your judgment in interpreting them when required, and keep a note of the assumptions you made.

1. **Branch Information:** The MA System shall keep information on each branch including its name and address, and the number of employees who work there. The system shall also contain information on which days (Mon-Sun) the branch is open, and opening hours. It will also keep information on opening hours (e.g., Mon-Fri 10:00AM-5:30PM; Sat 9:00AM-9:00PM; Sun Closed).
2. **Product Information:** The system shall contain relevant information on products of different types at the “item level”, such as: type (wine/beer/spirit/...), packaging info (can/bottle/...), volume (e.g., 375ml X 6 pack), price, and brand (e.g. Tooheys Old Dark Ale), as well as current stock level.
3. **Staff Information**: The system shall record information on staff members who work at different branches of MA. This will include their roles, type of employment (e.g. permanent, casual), salary (annual or hourly depending on permanent or casual), as well as who they report to.
4. **Membership Information**: The system shall record information on magic members, including type of membership (Platinum/Gold/Silver), and when the membership will expire.
5. **Sales Campaign Information.** The system shall keep information on sales campaign. Assume that these campaigns are global (same across all branches of MA). It will have information of the form: campaign start date and campaign end date, what items are on sale, and the discount for customers based on their membership (e.g., nonmembers 10%, Silver 15%, and Platinum/Gold 20%).