Please find out my response to the below in exercises.

Exercise 1:

Demonstrate your knowledge of asynchronous programming by creating a method that downloads three resources and aggregates the content length of all 3 responses. The caller should be able to cancel the operation at any time.

Notes / Assumptions:

* The method should be written as efficiently as possible.
* You can assume each resource you download is a string via a HTTP GET request.
* You can assume each resource exists. Eg. No error handling for HTTP responses.
* You can assume that each response returns all standard header you would expect from a normal HTTP rest request.

Exercise 2:

A coupon management system stores a definition of a coupon and every time a user redeems a

coupon. A coupon consists of an id, a title, a start date, end date, maximum number of coupons per user and also a maximum number of coupons across all users. When a user redeems a coupon the system keeps track of the users identifier, the datetime when the redemption occurred and a unique code that is generated by the application.

The coupon management system is required to provide the following capabilities:

* Provide an active list of coupons.
* Determine if a consumer can redeem a coupons.
* Store redemptions as they occur.
* Provide reporting on the redemptions for a specific offer.

The coupon management system is expected to have the following capacity:

* Coupons up to 50000 unique coupons. Generally, 100-500 active at any point in time.
* Redemptions – Upwards of 1 billion rows. Expect at least 1 million redemptions per day.

Requirements:

* Create a data model to represent the data in the above scenario. You are free to choose any storage mechanism you feel appropriate. The model should take into account the volume and expected operations into account.
* Define additional optimizations that you would apply to the basic data model. (For example, if you were to choose SQL you might want indexes on certain columns) Note:

Implementation/Code is not required.

**Solution 2:** Below is the data table structure to use to store data.

Table

Description automatically generated with medium confidence

**TBL-Coupons – holds coupons details**

**TBL-CouponRedeem – holds coupon redemptions information made by the user-**

* **Assuming there is trigger on this table ON INSERT operation that will update count of users redemptions and update RedeemCount of particular user in TBL-CustomerCouponRedeem.**

**Capacity and queries:**

* Provide an active list of coupons.
* This list can be retrieved by querying on TBL-Coupons by applying filtering on Coupon enddate (eg.where enddate>=currentdate)
* Determine if a consumer can redeem a coupons.
* There should be two conditions met before customer can apply for redemption of coupon. (Assuming coupon id – 1000, Customer id - 1 )
  + Total number of coupons redeemed (take count from TBLCouponRdeem) is less than TBlCoupon.MaxcouponCustomers for that coupond id – 1000
  + Total number of coupons redeemed by the user(take count for User id from TBL-CustomerCouponRedeem ) is less than TBLCoupon.MaxcouponperCustomer for coupond id – 1000 and user Id - 1
* Store redemptions as they occur.
* This will create a record and store in TBL-CouponRedeem whenever the customer redeem the coupon.
* Provide reporting on the redemptions for a specific offer.
* There are multiple reports can be generated as needed as below.
  + Number of redemptions made for a specific coupon.
  + Number of users made redemption for a coupon.
* For this kind of reports we can use in-built SQL reporting tool that can be used to create quick reports as needed for the defined schema.
* It does not require any query or stored procedure to create this reporting.
* We can implement ad-hoc reporting functionality that can be used recurringly on weekly, monthly, or specific time.

The coupon management system is expected to have the following capacity:

* Coupons up to 50000 unique coupons. Generally, 100-500 active at any point in time.
* It can be achieved by some background schedulers to create new unique coupons by checking above conditions
* Redemptions – Upwards of 1 billion rows. Expect at least 1 million redemptions per day.
* Based on considering this scenario we can also use NO-SQL database like. Azure Cosmos or MongoDB