**QUERIES FOR USER STORIES**

**User Story – GraphQL**

**User Story 1: Querying Product Data**

**Given I have a list of product IDs,  
 When I create a GraphQL query to fetch product details,  
 Then I should receive only the product name, price, and category fields.**

query GetProductDetails($productIds: [Int!]!) { products(where: { productID\_in: $productIds }) { productName price category { categoryName } } }

**User Story 2: Mutating User Profile Information**

**Given I need to update a user's email and address,  
 When I write a GraphQL mutation for updating user profile information,  
 Then the system should update the user details and return the updated information.**

mutation UpdateUserProfile($userId: Int!, $email: String!, $address: AddressInput!) { updateUser(id: $userId, email: $email, address: $address) { userID email address { houseNumber city state } } }

**User Story 3: Handling Complex Relationships**

**Given I need to retrieve an order with its items and associated product details,  
 When I use a GraphQL query to fetch this data,  
 Then I should receive a single response with all the nested information.**

query GetOrderWithItems($orderId: Int!) { order(id: $orderId) { orderID status orderItems { quantity price product { productName price } } } }

**User Story 4: Implementing GraphQL Subscriptions**

**Given an order's status may change,  
 When I implement a GraphQL subscription for order status updates,  
 Then I should receive real-time notifications when the order status changes.**

query GetOrderStatus($orderId: Int!) { order(id: $orderId) { orderID status } }

**User Story 5: Combining Multiple Data Sources**

**Given I want to see product reviews and ratings together,  
 When I use GraphQL to query data from both reviews and products,  
 Then I should receive a combined view of the product's feedback and ratings in a single response.**

query GetProductReviews($productId: Int!) { product(id: $productId) { productName price reviews { rating user { firstName lastName } } } }