**Order Flow Analysis**

**Objective**

* To analyse the order flow in Magento 2, specifically focusing on the checkout process, quoting, order placement with different product types, and the related tables and triggering events for invoices and credit memos.

**Checkout process**

* **Checkout Flow:**
  + Customer adds products to cart.
  + Proceeds to checkout.
  + Enters shipping and billing information.
  + Selects shipping method and payment method.
  + Places the order.
* **Related Tables:**
  + **quote**: Stores the current quote information.
  + **quote\_item**: Contains the items added to the quote.
  + **quote\_address**: Stores the shipping and billing addresses for the quote.
  + **quote\_payment**: Holds payment information for the quote.
  + **checkout\_session**: Tracks the current checkout session.

**Quote Management**

* **Quoting Process:**
  + Customer requests a quote for products.
  + Admin creates a quote with negotiated prices.
  + Quote is sent to the customer.
  + Customer can accept, reject, or negotiate the quote.
* **Related Tables:**
  + **quote**: Stores the quote information.
  + **quote\_item**: Contains the items in the quote.
  + **quote\_address**: Holds the addresses associated with the quote.
  + **quote\_address**: Stores the shipping and billing addresses for the quote.
  + **quote\_payment**: Holds payment information for the quote.
  + **checkout\_session**: Tracks the current checkout session.

**Order Placement with different Product Types:**

* **Order Placement:**
  + Customer selects products and adds them to the cart.
  + Proceeds to checkout.
  + Places the order with different product types (simple, configurable, virtual, etc.).

**Order Placement Process**

* **Customer adds products to cart**: When a customer adds products to the cart, Magento creates a quote (if not already created) and adds the selected items to the **quote\_item** table.
* **Customer proceeds to checkout**: At checkout, the customer provides shipping and billing information. These details are stored in the **quote\_address** table.
* **Order placement**: When the customer confirms the order, Magento converts the quote into an order. The quote data is copied to the **sales\_order** table, and the status of the quote changes to 'ordered'.
* **Inventory management**: If configured, Magento adjusts the inventory levels in the **cataloginventory\_stock\_item** table to reflect the products' sale.
* **Payment processing**: If payment is involved, payment-related data is stored in tables like **sales\_order\_payment**.
* **Order fulfillment**: Once the order is placed, it can be processed for shipping. Shipping information may be stored in tables like **sales\_shipment** and **sales\_shipment\_track**.
* **Related Tables:**
  + **sales\_order**: Stores the order information.
  + **sales\_order\_item**: Contains the items in the order.
  + **sales\_order\_address**: Holds the shipping and billing addresses for the order.

**Invoice and credit memo**

* **Invoice Generation:**
  + Order is shipped or completed.
  + Invoice is generated automatically or manually by the admin.
  + Invoice is sent to the customer.
* **Credit Memo Creation:**
  + Customer requests a refund or return.
  + Admin creates a credit memo.
  + Refund is processed.
* **Related Tables:**
  + **sales\_invoice**: Stores invoice information.
  + **sales\_invoice\_item**: Contains the items in the invoice.
  + **sales\_creditmemo**: Holds credit memo information.
  + **sales\_creditmemo\_item**: Contains the items in the credit memo.

**Triggering Events**

* **Invoice Cancellation Events:**
  + **sales\_order\_invoice\_cancel**: Triggered when an invoice is cancelled.
  + **sales\_order\_payment\_cancel\_invoice**: Triggered when a payment related to an invoice is cancelled.
  + **sales\_order\_invoice\_delete\_before**: This event is dispatched before an invoice is deleted. You can observe this event to perform actions before the invoice is deleted.
  + **sales\_order\_invoice\_delete\_after**: This event is dispatched after an invoice is deleted. You can observe this event to perform actions after the invoice is deleted.
  + **sales\_order\_invoice\_refund**: This event is dispatched when a refund is processed for an invoice. You can observe this event to perform actions when a refund is processed for an invoice.
  + **sales\_order\_invoice\_register**: This event is dispatched when an invoice is created for an order. It is triggered before the invoice is saved.
  + **sales\_order\_invoice\_save\_after**: This event is dispatched after an invoice is saved.
  + **sales\_order\_invoice\_pay**: This event is dispatched when an invoice is paid.
  + **sales\_order\_invoice\_process\_relation**: This event is triggered when an invoice is associated with an order. It occurs during the process of linking an invoice to an order. Developers might use this event to perform custom actions when an invoice is linked to an order, such as updating related data or sending notifications.
* **Credit Memo Cancellation Events:**
  + **sales\_order\_creditmemo\_cancel**: Triggered when a credit memo is cancelled.
  + **sales\_order\_payment\_cancel\_creditmemo**: Triggered when a payment related to a credit memo is cancelled.
  + **sales\_order\_creditmemo\_save\_before**: This event is dispatched before a credit memo is saved.
  + **sales\_order\_creditmemo\_save\_after**: This event is dispatched after a credit memo is saved.
  + **sales\_order\_creditmemo\_cancel**: This event is dispatched when a credit memo is canceled.
  + **sales\_order\_creditmemo\_refund**: This event is dispatched when a refund is processed for a credit memo. You can observe this event to perform actions when a refund is processed for a credit memo.
  + **sales\_order\_creditmemo\_register\_before**: This event is dispatched before a credit memo is registered.
  + **sales\_order\_creditmemo\_register\_after**: This event is dispatched after a credit memo is registered.
  + **sales\_order\_creditmemo\_item\_refund**: This event is dispatched when an item refund is processed for a credit memo.