

Odoo 18 Enterprise Sales MCQs with Answers & Real-time Explanations with examples

Module: Sales

Source: Google Gemini - 2.5 flash

Disclaimer: These MCQs are only for training purpose and to polish your functional knowledge, these are sample MCQs, please don't consider like same MCQS will be asked in Odoo Official Functional Certification Examination for any version

Follow me on LinkedIn: <https://www.linkedin.com/in/keriwala-sakib/>

Please let me know your suggestions and feedback after reading this document

I. Quotation & Sales Order Management

1. **Scenario:** Your company often sends out initial quotations that require customer review before a final proposal is generated. You want to ensure these initial quotes don't automatically reserve stock and can be easily distinguished from confirmed sales orders. Which Odoo Sales configuration setting or workflow best supports this requirement?
 - A) Set the "Quotation Validity" to 0 days.
 - B) Keep quotations in the "Quotation" state until manually confirmed by the sales team.
 - C) Enable "Automatic Sales Order Confirmation" in settings.
 - D) Configure a "Lock Confirmed Sales Orders" rule.

2. **Answer: B**

Explanation & Example: Odoo's default sales workflow starts with a "Quotation" state. In this state, products are generally *not* reserved in inventory (unless "Sales Warnings" are configured for insufficient stock or specific product routes are set). This allows for negotiation and changes without impacting stock. The confirmation of the quotation to a "Sales Order" is a manual action by the sales team, which then typically triggers inventory reservation and other downstream processes. **Example:** A sales rep creates a quotation for 50 units of "Product X." The inventory shows 40 units available. If the quotation is in the "Quotation" state, no stock is reserved. If the rep later confirms the quotation, Odoo will attempt to reserve 50 units, and if only 40 are available, it might generate a "Waiting for stock" message or a "delivery exception," depending on inventory settings. This separation prevents premature stock allocation.

3. **Question:** Your sales team frequently creates sales orders for services that are delivered over several months. You need to invoice the customer 50% upfront upon order confirmation and the remaining 50% upon completion of the service. How would you configure this invoicing policy in Odoo Sales?

- A) Set the "Invoicing Policy" on the sales order to "On Delivery."
- B) Create two separate sales orders, one for the down payment and one for the remaining balance.
- C) Use the "Milestone Invoicing" feature with down payments configured.
- D) Manually create two separate invoices from the sales order at different times.

4. **Answer: C**

Explanation & Example: Odoo's "**Milestone Invoicing**" (often combined with down payments or payment terms) is designed for this. You can create "Service" type products that represent milestones or a generic down payment product. When creating the sales order, you add a line for the down payment (e.g., 50% of the total). You then create an invoice for this down payment. Later, when the service is completed, you invoice the remaining balance. **Example:** For a "Consulting Project" worth \$10,000, you create a Sales Order. You add a line for "Consulting Project" (\$10,000) and then add a "Down Payment" line for 50%. You can then click "Create Invoice" on the sales order, and Odoo will ask if you want to invoice the down payment. After the first invoice is paid and the project is complete, you can again click "Create Invoice" to invoice the remaining 50%.

5. **Scenario:** A customer has confirmed a large sales order for 100 units of "Product A." Due to a temporary stock shortage, you can only deliver 70 units immediately. You want to deliver the remaining 30 units as soon as they are available without creating a new sales order. How does Odoo Sales and Inventory handle this situation by default?

- A) Odoo will block the entire delivery until all 100 units are available.
- B) Odoo will allow a partial delivery of 70 units and automatically create a "backorder" for the remaining 30 units.
- C) The sales order must be split into two separate orders manually.
- D) The sales order will automatically be cancelled.

6. **Answer: B**

Explanation & Example: By default, Odoo's Inventory module is configured to allow "**partial deliveries**" and automatically create "**backorders**" for the remaining quantities. This ensures that customers receive available products quickly while the system tracks pending deliveries. **Example:** When you process the delivery for the 100 units, and only 70 are available, you will validate the delivery for 70. Odoo will then ask if you want to create a backorder for the remaining 30. If you confirm, a new delivery order (or a line on the existing one) will be created for the 30 units, automatically linked to the original sales order, which remains in a "Waiting" state until the backorder is delivered.

7. **Question:** Your sales team uses standardized terms and conditions, along with specific product bundles, for certain types of sales. To ensure consistency and speed up quotation creation, what Odoo feature should be utilized?

- A) "Product Categories" to group products.

- B) "Quotation Templates" with pre-defined products, optional products, and terms & conditions.
- C) "Automated Actions" to add standard lines.
- D) "Price Lists" to apply standard pricing.

8. **Answer: B**

Explanation & Example: "Quotation Templates" are specifically designed for this purpose. They allow you to pre-define:

- Standard product lines.
- Optional products for upselling/cross-selling.
- Specific payment terms.
- Terms and conditions.
- Even confirmation emails. This significantly reduces manual entry and ensures compliance with sales policies. **Example:** You create a "Standard Software Package" quotation template. It includes "Software License Basic" and "Installation Service." In the "Optional Products" tab, you add "Premium Support." In the "Terms & Conditions" tab, you input your standard legal text. When a sales rep creates a new quotation, they simply select this template, and all these items are pre-filled, ready to be sent or customized.

II. Product & Pricing

5. **Scenario:** Your company sells software licenses (main product) and accompanying support plans (optional extra). The support plan price depends on the number of licenses sold (e.g., 10% of license value for quantities up to 100, 8% for quantities over 100). How can Odoo's pricing mechanism effectively manage this dynamic pricing structure?

- A) Manually adjust the support plan price on each sales order.
- B) Create multiple product variants for the support plan.
- C) Use "Price Lists" with "Advanced Price Rules" based on quantity and percentage.
- D) Configure "Sales Warnings" for manual price calculation.

6. **Answer: C**

Explanation & Example: Odoo's "Price Lists" with "Advanced Price Rules" are the most robust solution for complex pricing scenarios. You can define rules using formulas, percentages, fixed amounts, and minimum quantities. **Example:** You create a pricelist named "Standard Customer Pricing." Within this pricelist, you add price rules for your "Software Support Plan" product:

- Rule 1: If Min. Quantity is 1, Computation: "Formula", Based On: "Sales Price" (of the license), Discount: "-10%" (meaning 10% of the base price). This effectively sets the support price to 10% of the license price.
- Rule 2: If Min. Quantity is 101, Computation: "Formula", Based On: "Sales Price", Discount: "-8%". When the sales rep adds the software license and

then the support plan to the sales order, Odoo automatically applies the correct percentage based on the quantity of licenses.

7. **Question:** Your company has different pricing tiers for customers based on their volume and relationship (e.g., "Wholesale," "Retail," "Premium"). How would you ensure that a customer automatically gets assigned the correct price list when a sales order is created for them?
- A) Manually select the price list on each sales order.
 - B) Assign the default "Pricelist" directly on the customer's contact record.
 - C) Set a "Sales Team" for each price tier.
 - D) Create a separate "Quotation Template" for each price tier.

8. **Answer: B**

Explanation & Example: In Odoo, the most straightforward way to link a customer to a specific pricing tier is by assigning a **default "Pricelist" directly on their contact record** (Partner form). When a new sales order is created for that customer, Odoo will automatically apply their assigned pricelist. **Example:** For "ABC Corp," you open their contact record in Odoo and set their "Pricelist" field to "Wholesale Pricing." Now, every time a new sales order is created for ABC Corp, the "Wholesale Pricing" pricelist will be automatically selected, and all product prices will be calculated according to its rules.

9. **Scenario:** You have a product, "Customizable Widget," which comes in various colors (Red, Blue, Green) and materials (Plastic, Metal). These are distinct options that affect pricing and inventory. How should you set up this product in Odoo to manage all combinations efficiently?
- A) Create a separate product for each combination (e.g., "Red Plastic Widget," "Blue Metal Widget").
 - B) Use a single product with "Product Variants" defined by "Attributes" (Color, Material).
 - C) Define each combination as an "Optional Product."
 - D) Use "Tags" to differentiate the variations.

10. **Answer: B**

Explanation & Example: Odoo's "**Product Variants**" feature is precisely for this. You define "Attributes" (like "Color" and "Material") and their "Values" (like "Red," "Blue," "Green" for Color; "Plastic," "Metal" for Material). Odoo then automatically generates all possible combinations as product variants. Each variant can have its own price, internal reference, barcode, and be tracked independently in inventory.

Example: You create a product "Customizable Widget."

- Add Attribute: "Color" with values "Red," "Blue," "Green."
- Add Attribute: "Material" with values "Plastic," "Metal." Odoo will then generate variants like "Customizable Widget (Red, Plastic)," "Customizable Widget (Red, Metal)," etc., totaling 6 variants. Each of these variants will have its own stock count and can be managed individually.

III. Invoicing & Payments

8. **Question:** A sales order has been confirmed, and the customer has made a partial payment. You need to record this payment and then invoice the remaining balance. What is the correct Odoo workflow for handling partial payments and subsequent invoicing?

- A) Manually create a partial invoice and then a full invoice.
- B) Register the payment on the sales order, which then allows invoicing the remaining amount.
- C) The sales order must be fully invoiced before any payment can be registered.
- D) Partial payments are managed in the Accounting module only, not directly from Sales.

9. **Answer: B**

Explanation & Example: Odoo integrates sales, accounting, and payments. You can **register payments directly on the sales order** or its associated draft invoice. If a payment is registered on a sales order that isn't fully invoiced, it creates an "Outstanding Credit" for the customer. When you then create an invoice for the remaining amount, Odoo will prompt you to reconcile the outstanding credit, effectively applying the partial payment. **Example:** A sales order for \$1,000 is confirmed. The customer pays \$300 upfront. You click "Register Payment" on the Sales Order and input \$300. This creates a journal entry and marks the sales order as partially paid. Later, when you create the invoice for \$1,000, Odoo will show an "Outstanding Credits" banner, and you can apply the \$300 payment to reduce the invoice balance to \$700.

10. **Scenario:** Your accounting department requires that all invoices generated from sales orders use a specific journal (e.g., "Sales Invoices - Online"). How can you ensure this journal is automatically selected when sales orders are confirmed and converted to invoices?

- A) Manually select the journal on each invoice.
- B) Configure the default "Sales Journal" in the Sales module settings.
- C) Assign the specific journal to the "Sales Team" responsible for the sales order.
- D) Set the "Invoicing Journal" on the "Quotation Template" used for the sales order.

11. **Answer: D**

Explanation & Example: The "**Quotation Template**" (also known as Sales Order Template) allows you to define a specific "Invoicing Journal." When a sales order is created using that template and then converted to an invoice, it will automatically use the journal specified in the template. If no journal is set on the template, it defaults to the sales journal with the lowest sequence. **Example:** You have a "Website Orders" quotation template. You configure this template to use the "Sales Invoices - Online" journal. Now, every time a sales order is generated from your website (which uses

this template) and subsequently invoiced, the invoice will automatically be created in the "Sales Invoices - Online" journal, streamlining accounting processes.

IV. Delivery & Logistics Integration

10. **Question:** Your warehouse needs to prepare shipments for sales orders. When a sales order for physical products is confirmed, what Odoo module is automatically engaged, and what document is typically created to manage the delivery process?

- A) The Project module, creating a new task.
- B) The Manufacturing module, creating a new production order.
- C) The Inventory module, creating a "Delivery Order" (picking operation).
- D) The Purchase module, creating a new Request for Quotation.

11. **Answer: C**

Explanation & Example: This is a core integration between Sales and Inventory. When a sales order for storable products is confirmed, Odoo automatically creates a **"Delivery Order"** (or a "Picking" operation, depending on your warehouse flow configuration) in the **Inventory module**. This document guides the warehouse staff on which products to pick, pack, and ship. **Example:** A sales order for 5 units of "Laptop Model X" is confirmed. Immediately, a "Delivery Order" for those 5 laptops is created in the Inventory module, typically in the "Ready" state if stock is available. Warehouse staff can then open this delivery order, pick the items, validate the transfer, and eventually mark it as "Done," indicating shipment.

12. **Scenario:** Your company allows customers to choose their preferred shipping method during the sales process (e.g., Standard, Express, Customer Pickup). You need to ensure that the chosen shipping method is reflected on the sales order and potentially affects the shipping cost calculated. How do you configure this in Odoo?

- A) Add a custom text field for "Shipping Notes" on the sales order.
- B) Use the "Delivery Methods" feature in the Sales module settings.
- C) Create different "Product Variants" for each shipping option.
- D) Manually add shipping lines to the sales order.

13. **Answer: B**

Explanation & Example: Odoo's **"Delivery Methods"** feature (also known as Shipping Methods) allows you to define various shipping carriers and their associated pricing rules (e.g., flat rate, by weight, by quantity). Once configured, these methods can be selected on the sales order, and Odoo will calculate the shipping cost based on the method and the order's contents. **Example:** You configure a "Standard Shipping" delivery method (flat rate \$10) and an "Express Shipping" method (calculated by weight). When a sales rep creates a sales order, they can select one of these methods. If "Express Shipping" is chosen, Odoo automatically adds a shipping line to the sales order with the calculated cost based on the products' weight, integrating directly with the sales order total.

V. Reporting & Analytics

12. **Question:** Your sales manager wants to analyze the profitability of sales by product category over the last quarter. Which Odoo Sales report would provide this information most effectively, and what "Measures" would be crucial to include?

- A) "Salesperson Leaderboard" focusing on "Total Sales."
- B) "Sales Analysis" report (Pivot Table) grouped by "Product Category" and using "Margin" as a measure.
- C) "Activities Report" filtered by sales orders.
- D) "Inventory Valuation" report.

13. **Answer: B**

Explanation & Example: The **"Sales Analysis" report** (found under [Sales > Reporting > Sales](#)) is the primary tool for detailed sales analytics. Using its **Pivot Table** view, you can group data by dimensions like "Product Category," "Sales Team," "Customer," or "Salesperson." By selecting **"Margin"** (which represents $\text{Sales Price} - \text{Cost Price}$) as a measure, you can directly assess profitability.

Example: The sales manager opens the Sales Analysis report, switches to Pivot view, drags "Product Category" to the Rows, and "Order Date (Quarter)" to the Columns. They then select "Margin" as the Measure. The resulting table will show the total margin generated by each product category for each quarter, allowing for direct comparison of profitability.

14. **Scenario:** You've implemented an upselling strategy where you suggest "Optional Products" on your quotations. Your sales director wants to track the effectiveness of this strategy: how often are optional products added to confirmed sales orders? Which report or analysis method would you use in Odoo?

- A) "Sales Order Line Analysis" grouped by "Optional Product" status.
- B) Reviewing individual sales orders and counting manually.
- C) A custom "Sales Analysis" report with a filter for optional products that were actually sold.
- D) Odoo doesn't track the conversion of optional products.

15. **Answer: C**

Explanation & Example: While Odoo's standard Sales Analysis reports are powerful, for very specific conversion metrics like "Optional Product Conversion," you often leverage the **Sales Analysis pivot table with specific filters and groupings**. You'd typically look for sales order lines that originated as optional products on a quotation but were included in the final confirmed sales order. This might involve creating a custom field via Studio or using the "Is Optional" field on the sales order line (which exists on the quotation template side). **Example:** You go to [Sales > Reporting > Sales](#) (or [Sales > Reporting > Orders](#) if available for lines) and switch to the Pivot View. You might need to add a filter for "Sales Order Line Type" (if categorized) or "Product Is Optional" (if that field is exposed/customized). Then, you'd measure "Quantity" or "Untaxed Amount" and group by "Product" or

"Date" to see which optional products are being converted and at what rate. Advanced analysis might require Odoo.sh or custom reporting tools if the standard UI doesn't expose the exact historical optional product status.

VI. Configuration & Customization

14. **Question:** Your sales team has specific information they need to capture on each sales order, such as the "Customer's PO Number" and "Expected Delivery Date" (beyond the standard scheduling). These fields are critical for order fulfillment and internal tracking. How would you add these fields to the sales order form in Odoo 18 without writing code?

- A) Manually add them as notes in the Chatter.
- B) Utilize Odoo Studio to create and add custom fields to the Sales Order model.
- C) Ask a developer to create a new module.
- D) This functionality is not possible in Odoo's standard Sales module.

15. **Answer: B**

Explanation & Example: Odoo Studio is the primary tool in Odoo Enterprise for no-code/low-code customization. It allows administrators or functional consultants to add custom fields, modify views, create automated actions, and reports directly from the user interface. **Example:** You open a Sales Order form, activate Odoo Studio, and drag a "Text" field onto the form for "Customer PO Number" and a "Date" field for "Expected Delivery Date." Once saved, these fields become part of the Sales Order model, are visible on all sales orders, and can be used for filtering, grouping, and reporting.

16. **Scenario:** To improve sales efficiency, you want to automate the assignment of sales orders. Specifically, if a sales order is created for a customer located in "Germany," it should automatically be assigned to the "DACH Sales Team." How can you configure this automation?

- A) Manually assign sales orders to the correct team.
- B) Set up an "Automated Action" (Server Action) on the Sales Order model.
- C) Use a "Sales Team" rule in the CRM module.
- D) Implement a custom Python script.

17. **Answer: B**

Explanation & Example: "Automated Actions" (also known as Server Actions) in Odoo allow you to define rules that trigger actions based on specific conditions on a record. This is a powerful no-code automation feature. **Example:**

- **Model:** Sales Order (`sale.order`)
- **Trigger:** On Creation (or On Create & Update if customer info might change)
- **Conditions:** `Customer.Country` is `Germany`

- **Action:** "Update Record" -> Set **Sales Team** to **DACH Sales Team** Now, any sales order created for a customer with "Germany" as their country will automatically be assigned to the DACH Sales Team, ensuring proper team allocation and reporting.

VII. Advanced Features & Integrations

16. **Question:** Your company is piloting a new "Upselling" initiative. On every confirmed sales order for Product A, you want to suggest Product B (a higher-value accessory) to the customer via an email, encouraging them to buy it as an add-on. How can Odoo support this process effectively?

- A) Manually track Product A sales and send emails for Product B.
- B) Use the "Optional Products" feature on Product A's sales order line to suggest Product B.
- C) Configure an "Automated Action" on Sales Order confirmation to send a targeted email with a link to Product B.
- D) Odoo's Sales module does not have dedicated upselling features for post-confirmation.

17. **Answer: C**

Explanation & Example: While "Optional Products" are for pre-confirmation upselling on the quote, post-confirmation upselling (or cross-selling) often leverages **"Automated Actions"** combined with **"Email Templates."** This allows you to trigger specific actions and communications after a certain stage in the sales process.

Example:

- **Model:** Sales Order (**sale.order**)
- **Trigger:** On Update (or On Confirmation)
- **Conditions:** **State** is **Sales Order** AND **Product A** is in **order lines**
- **Action:** "Send Email" using a predefined email template (e.g., "Exclusive Offer for Product B"). The email template can include a link to Product B on your website or a contact person. This automated email encourages further purchases from customers who have already committed to a main product.

18. **Scenario:** Your sales team sometimes needs to issue pro-forma invoices to customers as a preliminary document before the actual commercial invoice. How can you generate a pro-forma invoice in Odoo Sales without affecting the final accounting entries?

- A) Create a standard invoice and mark it as "Draft."
- B) Utilize the "Pro-forma Invoice" option available on the sales order.
- C) Manually create a PDF document outside of Odoo.
- D) Pro-forma invoices are not supported in Odoo.

19. **Answer: B**

Explanation & Example: Odoo provides a specific feature for **"Pro-forma Invoices"** directly from the sales order. This generates a document that looks like an invoice but does not create any accounting entries (it's not journalized). It serves as a

commitment or estimation before the formal invoice is issued. **Example:** A customer requests a pro-forma invoice for a confirmed sales order before they can process payment. The sales rep navigates to the sales order, clicks the "Print" button (or a dedicated "Pro-forma" smart button/action if configured), and selects "Pro-forma Invoice." Odoo generates a PDF document that the rep can send to the customer, clearly labeled as "Pro-forma."

20. **Question:** Your company integrates Odoo Sales with an external e-commerce platform. When a customer places an order on the e-commerce site, how does Odoo typically handle the creation of a sales order, and what is a common field used to link these external orders to a specific Odoo sales team?

- A) Orders are imported as leads first, then converted to sales orders.
- B) E-commerce orders are automatically created as sales orders in Odoo, often linked to a specific "Sales Team" configured for the e-commerce channel.
- C) Manual data entry is required for all e-commerce orders.
- D) E-commerce orders are only visible in the Inventory module.

21. **Answer: B**

Explanation & Example: Odoo's e-commerce module (or direct integrations via APIs for external platforms) is designed for seamless order creation. When an order is placed online, it automatically generates a **sales order in Odoo**, usually associated with a specific **"Sales Team"** (e.g., "Online Sales Team") and a "Website" linked to the e-commerce configuration. **Example:** In **Website > Configuration > Settings > Shop - Checkout Process**, you can select a "Sales Team" (e.g., "E-commerce Team") and a "Salesperson" to handle web orders. Any order placed through your Odoo e-commerce will automatically populate the sales order with this team and salesperson, streamlining the backend processing.

VIII. Multi-Company & Multi-Currency

19. **Scenario:** Your Odoo instance manages sales for two different legal entities, "Company A" (operating in USD) and "Company B" (operating in EUR). A customer from "Company A" places an order for a product priced in JPY. How does Odoo Sales manage the currency and revenue reporting for this scenario?

- A) The sales order must be created in the customer's currency (JPY), but the Expected Revenue will be reported in Company A's default currency (USD) using configured exchange rates.
- B) All sales orders must be created in the company's default currency (USD).
- C) Odoo will automatically convert JPY to EUR, then to USD.
- D) Multi-currency sales require manual currency conversion on each order line.

20. **Answer: A**

Explanation & Example: Odoo's **multi-currency capabilities** are robust. You can

set the currency of a sales order to be the customer's currency (JPY in this case). Odoo will record the amounts in JPY on the sales order. However, for internal reporting (e.g., Sales Analysis reports, Profit & Loss statements), Odoo will convert these JPY amounts to the sales order's associated company's default currency (USD for Company A) using the exchange rates configured in the Accounting module.

Example: A sales order is created for a Japanese customer in JPY. The total is JPY 1,000,000. When viewing the Sales Analysis report for "Company A," this JPY 1,000,000 will be converted to its USD equivalent (e.g., \$6,500) based on the exchange rate on the order date or the reporting date, ensuring consistent financial reporting across the organization.

21. **Question:** In a multi-company Odoo setup, a sales representative assigned to "Company X" needs to create a sales order for a customer who is serviced by "Company Y." What is the primary method to ensure this sales order is correctly linked to "Company Y" while the sales rep is working within "Company X's" environment?
- A) The sales rep must log out of "Company X" and log into "Company Y."
 - B) The sales order should be manually transferred between companies after creation.
 - C) The sales rep can change their active "Company" selector in the Odoo top bar before creating the sales order.
 - D) Multi-company sales are not supported for a single sales representative.

22. **Answer: C**

Explanation & Example: Odoo allows users in a **multi-company environment** to switch their active company from the top bar. This changes the context of their work, meaning any new records created (like sales orders) will be associated with the currently selected company. **Example:** Sarah is a sales rep primarily working for "Company X." She receives a request from a customer that needs to be billed by "Company Y." Before creating the sales order, Sarah clicks the "Company" selector in the top right of the Odoo interface (next to her username) and switches to "Company Y." She then creates the sales order. This sales order will automatically be linked to "Company Y" and its associated settings (like pricing, warehouses, accounting).