Odoo 18 Enterprise Purchase MCQs with Balanced Answers & Real-time Explanations

Module: Purchase
Source: Google Gemini - 2.5 flash
Disclaimer: These MCQs are only for training purposes and to polish your functional knowledge. These are sample MCQs, please don’t consider that the 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. Request for Quotation (RFQ) & Purchase Order (PO) Management

1. **Scenario:** Your company requires that all purchase orders above a certain value (e.g., $10,000) must be approved by the Purchasing Manager before they can be sent to the vendor or confirmed. Which Odoo feature would you use to implement this approval workflow without custom code?

   1. A) Set the "Responsible" user on the Purchase Order to the Purchasing Manager.
   2. B) Configure "Purchase Agreement" of type "Blanket Order" for high-value purchases.
   3. C) Implement an "Approval Rule" (via settings or a specific app like 'Purchase Order Approval').
   4. D) Manually send an email to the Purchasing Manager for confirmation.
2. **Answer: C**

   Explanation & Example: Odoo Enterprise often provides built-in **"Approval Rules"** within modules like Purchase (or dedicated approval apps) to manage such workflows. These rules can be configured based on criteria like order total, product category, or specific user roles. When a PO meets the defined criteria, it moves into a "To Approve" state, requiring approval from designated users before proceeding. **Example:**

   1. Go to Purchase > Configuration > Settings.
   2. Look for "Purchase Order Approval" settings or a dedicated Approvals app.
   3. Configure a new rule: If Total Amount is greater than $10,000, then Required Approvers is Purchasing Manager Group. Now, when a purchase agent creates a PO for $12,000 and tries to "Confirm

Order," the PO will move to a "To Approve" state, and the Purchasing Manager will receive a notification to review and approve it.

3. **Question:** Your vendor "Global Supplies" frequently updates prices for "Component X." To ensure your team always gets the most current pricing without manually checking vendor websites, where in Odoo would you link "Component X" to "Global Supplies" and specify their dynamic pricing rules (e.g., quantity-based tiers)?

    1. A) On the "Contacts" form of "Global Supplies."
    2. B) In the "Price Lists" module under "Supplier Price Lists."
    3. C) On the "Product" form of "Component X" within the Purchase tab.
    4. D) In the "Purchase Agreements" for "Global Supplies."
4. **Answer: C**

    Explanation & Example: The most direct and standard place to define vendor-specific pricing for a product in Odoo is on the **Product form itself, under the Purchase tab**. Here, you can add multiple vendor lines for the same product, specifying vendor name, their product code, specific price, minimum quantity to qualify for that price, and lead time. This allows Odoo to automatically suggest the correct price when an RFQ is created for that product from that vendor. **Example:**

    1. Go to Inventory > Products > Component X.
    2. Navigate to the Purchase tab.
    3. Click Add a line.
    4. Select Vendor: Global Supplies.
    5. Enter Vendor Product Name/Code (e.g., "CPNT-X").
    6. Set Price to $50, Quantity 1 (for base price).
    7. Add another line for Global Supplies: Price $45, Quantity 100 (for bulk discount). Now, when an RFQ is created for Component X and Global Supplies is selected, Odoo will automatically apply the $45 price if the quantity ordered is 100 or more.

II. Product Procurement & Automation

3. **Scenario:** Your warehouse team needs to replenish "Safety Gloves." The reordering rule for Safety Gloves is set to Min. Quantity 50, Max. Quantity 100. Current stock is 40 units. When the Odoo scheduler runs, how many units will it suggest purchasing for Safety Gloves?

    1. A) 50 units (to reach the minimum).
    2. B) 40 units (current stock - minimum).
    3. C) 60 units (to reach the maximum).

   4. D) 10 units (maximum - current stock).

4. **Answer: C**

   Explanation & Example: Odoo's "Reordering Rules" aim to bring the stock level *up to the maximum quantity* when the virtual stock falls below the minimum. The calculation is Quantity to Order = Max Quantity - Current Virtual Quantity. In this case, 100 (Max) - 40 (Current) = 60 units. **Example:**

   1. Product: Safety Gloves, Reordering Rule: Min=50, Max=100.
   2. Current On Hand stock: 40.
   3. When the Inventory > Operations > Run Scheduler is executed, Odoo detects 40 < 50.
   4. It generates a draft Request for Quotation for 100 - 40 = 60 units of Safety Gloves. This ensures enough stock to last until the next reorder point.

5. **Question:** Your company produces custom furniture. When a sales order is confirmed for a specific "Custom Sofa," its unique "Upholstery Fabric" (which is a component) needs to be ordered directly from a supplier *only* for that specific customer's order, without keeping it in general stock. What procurement route should be configured for "Upholstery Fabric"?

   1. A) The "MTS" (Make-to-Stock) route.
   2. B) A "Dropshipping" route.
   3. C) The "Buy" route with "Make-to-Order" enabled.
   4. D) A "Reordering Rule" set to zero for minimum stock.

6. **Answer: C**

   Explanation & Example: The **"Buy" route with "Make-to-Order" (MTO) enabled** on the product (or Bill of Material component) is the perfect solution for this "just-in-time" procurement. It ensures that a purchase order is generated *only when there is a specific demand* (like from a sales order or manufacturing order), preventing stock accumulation for unique or expensive components. **Example:**

   1. Open Product: Upholstery Fabric.
   2. Go to the Inventory tab.
   3. Ensure the Buy route is checked.
   4. Click on the Buy route details (or access Inventory > Configuration > Routes).
   5. Ensure the rule for Buy has "Procure from: Vendor" and "Supply Method: Create PO" and importantly, that it respects the **"Make to Order"** condition. Now, when a Sales Order for "Custom Sofa" is confirmed, and "Upholstery Fabric" is part of its BoM, Odoo will

automatically create an RFQ for the exact quantity of "Upholstery Fabric" needed for that specific sofa.

III. Receipts & Inventory Integration

5. **Scenario:** Your warehouse receives a shipment from a vendor containing "Product A" (10 units) and "Product B" (15 units) as per Purchase Order PO007. However, upon inspection, 2 units of "Product A" are found damaged and cannot be accepted. How should the warehouse team correctly record this partial receipt and damage in Odoo?

    1. A) Validate the receipt for the full quantities (10 of A, 15 of B) and then create a separate scrap order for the 2 damaged units of A.
    2. B) Adjust the "Done" quantity for "Product A" to 8 on the receipt, and then create a "Return" to the vendor for the damaged units.
    3. C) Adjust the "Done" quantity for "Product A" to 8 on the receipt and proceed with validating, creating a backorder for the remaining 2 units (if expected), or cancelling the remaining 2.
    4. D) Refuse the entire shipment and request a new one from the vendor.
6. **Answer: C**

    Explanation & Example: The most accurate way to handle this in Odoo is to record the actual received quantity and then decide on the remaining. By adjusting the "Done" quantity on the receipt to 8 for "Product A," you are telling Odoo what *actually* came in. Odoo will then typically prompt you to either create a backorder for the remaining 2 units (if you expect them later) or mark them as "No Backorder" if they won't be sent. Damaged items are usually a separate process (scrap or return, depending on policy). **Example:**

    1. Open the Receipt linked to PO007.
    2. On the line for Product A, change the Done quantity from 10 to **8**.
    3. Click Validate.
    4. Odoo will pop up a dialog asking: "Some products have not been moved. Do you want to create a backorder?" Select "No Backorder" if you consider the damaged 2 units not coming, or "Create Backorder" if the vendor will replace them. This correctly updates your stock for the received items and closes the specific line for those 2 units. A separate scrap or return could then be processed for the damaged items if needed.

IV. Vendor Bill & Accounting Integration

6. **Question:** After receiving goods from Vendor X, you go to create the vendor bill. The purchase order was for 100 units, but you only received

95 units, and the invoicing policy is "On Received Quantities." What quantity will Odoo automatically propose on the vendor bill line for this product?

1. A) 100 units.
2. B) 95 units.
3. C) 0 units, as the PO is not fully completed.
4. D) The system will not allow you to create a bill.

7. **Answer: B**

   Explanation & Example: When the **"Invoicing Policy" is set to "On Received Quantities"**, Odoo automatically proposes the quantity that has *actually been received* and validated in your inventory. This is a critical control to ensure you only get billed for what you have physically in your possession. **Example:**

   1. Purchase Order confirmed for 100 units of "Product Z".
   2. Partial receipt validated for 95 units of "Product Z".
   3. Go to the Purchase Order and click the Create Bill smart button.
   4. Odoo will automatically generate a draft vendor bill. On the line for "Product Z," the Quantity field will pre-fill with **95** units, reflecting the received quantity.

8. **Scenario:** You have a multi-company Odoo setup. "Company A" (operating in USD) purchases services from "Company B" (operating in EUR) which is also managed within the same Odoo instance. How would you ensure an inter-company purchase order is correctly generated in "Company A" to reflect a vendor bill from "Company B"?

   1. A) Manually create a Purchase Order in "Company A" and a Sales Order in "Company B."
   2. B) Configure "Inter-Company Flows" between "Company A" and "Company B" so that a PO in one automatically creates a SO in the other.
   3. C) Create a "Blanket Order" between the two companies.
   4. D) Assign a specific "Sales Team" for inter-company transactions.

9. **Answer: B**

   Explanation & Example: Odoo's **"Inter-Company Flows"** feature is specifically designed to automate transactions between entities within the same Odoo database. When configured, if Company A creates a Purchase Order for Company B, Odoo can automatically generate a corresponding Sales Order in Company B's environment, streamlining the internal billing process. **Example:**

   1. Go to Settings > General Settings > Companies.

   2. Select Company A and Company B.
   3. Under Inter-Company Flow, configure the "Create Sale Orders from Purchase Orders" and "Create Purchase Orders from Sale Orders" options between these two companies. Now, when a purchasing agent in Company A creates a PO for services from Company B, upon confirmation, Odoo automatically creates a Sales Order in Company B's database, ready for Company B to invoice Company A.

V. Reporting & Analytics

8. **Question:** Your purchasing department wants to assess the performance of your suppliers based on their average lead time (delivery speed). Which Odoo Purchase report or analysis tool would you use, and what measure would be crucial to include?

   1. A) "Vendor Bills" report, looking at due dates.
   2. B) "Purchase Analysis" report (Pivot Table), with "Average Delivery Days" as a measure.
   3. C) "Product Moves" report, filtered by vendor.
   4. D) "Inventory Valuation" report, filtered by purchase date.
9. **Answer: B**

   Explanation & Example: The **"Purchase Analysis" report** (found under Purchase > Reporting > Purchase Analysis) is the central hub for purchase-related analytics. In its **Pivot Table** view, you can select various measures. **"Average Delivery Days"** (or a similar lead time measure) is a standard measure for supplier performance related to speed. You would group by "Vendor" to see their individual performance. **Example:**

   1. Go to Purchase > Reporting > Purchase Analysis.
   2. Switch to Pivot view.
   3. Drag Vendor to the Rows.
   4. Add Average Delivery Days (or Average Days to Receive) to the Measures. This will generate a table showing each vendor and their average delivery time based on past purchase orders, allowing the purchasing manager to easily compare supplier speed.
10. **Scenario:** A specific product, "Specialized Gear," has seen unusually high purchase costs in the last quarter. You need to quickly identify all purchase orders for "Specialized Gear" from the last 3 months, review the exact unit prices paid, and see which buyers made those purchases. How would you best achieve this in Odoo?

    1. A) Manually go through all confirmed purchase orders for the last 3 months.

2. B) Use the "Purchase Order Lines" list view, applying filters for the product, date range, and then adding "Unit Price" and "Buyer" (Purchaser) columns.
3. C) Run the "Purchase Analysis" report, grouping by product and buyer.
4. D) Check the "Product History" in the Inventory module.

11. **Answer: B**

Explanation & Example: While the "Purchase Analysis" report (Option C) is good for aggregated data, to see *exact individual unit prices paid on specific POs* and link them directly to the buyer from a list, the **"Purchase Order Lines" list view** is more effective. You can easily filter, group, and add/remove columns to get the precise transactional detail needed.
Example:

1. Go to Purchase > Orders > Purchase Order Lines.
2. Apply Filters: Add Product is Specialized Gear. Add Order Date is in the last 3 months.
3. Click the "Measures" icon (often a small gear or three dots) on the list view header.
4. Select Unit Price and Purchaser (or Responsible if that's the buyer field). This will display a list of every line item for "Specialized Gear" in the last 3 months, showing the exact price paid on each order and the buyer responsible, allowing for quick review and drill-down into specific POs.

VI. Advanced Features

10. **Question:** Your company occasionally procures services that require significant upfront planning and phased payments. You want to manage these as purchase orders, but ensure that vendor bills are only generated when specific milestones are achieved (e.g., 25% upon contract signing, 50% upon service commencement, 25% upon completion). How can Odoo handle this invoicing method for purchases?

1. A) Set the "Invoicing Policy" to "On Ordered Quantities" and manually adjust the bill.
2. B) Create a separate purchase order for each milestone.
3. C) Use a "Service Product" configured for "Milestone Invoicing" combined with a purchase order.
4. D) Odoo does not support milestone-based vendor billing.

11. **Answer: C**

Explanation & Example: Odoo supports **"Milestone Invoicing"** for services, and this applies to both sales and purchases. You'd typically define a "Service" type product (or a product category) with the "Invoicing Policy" set appropriately (often "Based on Timesheets" or similar for services, but the *actual* invoicing is triggered by adding "Milestone" lines to the PO/SO). The key is to add the milestones as separate lines on the PO, or have a generic "down payment" product, and then invoice those specific lines as their conditions are met. **Example:**

1. Create a Product called "Project Phase 1 Completion" (type Service, or similar).
2. Create a Purchase Order for your Consulting Project (total amount).
3. Add product lines for each milestone, e.g., "Consulting Project - Phase 1" (25% of value), "Consulting Project - Phase 2" (50% of value).
4. When Phase 1 is complete, you can go to the PO and click Create Bill. Odoo allows you to select which lines you want to bill. You select "Consulting Project - Phase 1," create the bill, and validate it. This gives you precise control over partial billing based on service progress.

12. **Scenario:** Your purchasing department needs to regularly purchase items in bulk from a specific vendor, but the exact quantity is not known upfront. They want to issue a framework agreement that commits to a price for a certain period, allowing them to release specific purchase orders as needed against this agreement. Which type of "Purchase Agreement" is most suitable?

    1. A) A "Call for Tender" (Bid).
    2. B) A "Purchase Order" with multiple lines.
    3. C) A "Blanket Order" (Framework Agreement).
    4. D) A "Vendor Price List."

13. **Answer: C**

    Explanation & Example: The **"Blanket Order"** (or Framework Agreement) type of Purchase Agreement is precisely designed for such scenarios. It establishes a long-term contract with a vendor for specific products at agreed prices for a defined period, but without a fixed total quantity commitment at the outset. Individual purchase orders are then created "against" this blanket order, drawing from the agreed terms as quantities are needed. **Example:**

    1. Go to Purchase > Orders > Purchase Agreements.
    2. Click Create.
    3. Select Agreement Type: Blanket Order.
    4. Choose your Vendor.

5. Add the Products (e.g., "Office Paper") and Unit Price. You don't need to specify a total quantity at this stage for the overall blanket order, but rather an expected quantity over time if desired.
6. Set the Agreement Deadlines.
7. Confirm the Blanket Order. Now, any purchasing agent can go to this confirmed Blanket Order and click New Purchase Order. They select the specific quantity of "Office Paper" needed for that particular instance (e.g., 100 reams), and Odoo generates a new PO linked to the Blanket Order, automatically applying the agreed price.

Additional Odoo 18 Enterprise Purchase MCQs

Module: Purchase

---

1. **Scenario:** Your company implements a dropshipping strategy for certain products. When a customer places an order for "Gadget X" (which you dropship), a purchase order is automatically created and sent to your supplier, "Dropship Corp." The supplier then ships "Gadget X" directly to your customer. Which configuration on "Gadget X" is essential to enable this automated dropshipping flow via the Purchase module?

    1. A) Setting up a "Reordering Rule" with a low minimum quantity.
    2. B) Configuring the "Buy" route on the product and setting up "Dropship" as the fulfillment method.
    3. C) Creating a "Purchase Agreement" with "Dropship Corp."
    4. D) Manually creating a purchase order for each sales order and selecting the customer's address as the delivery address.
2. **Answer: B**

    Explanation & Example: Odoo's **Dropshipping** functionality is managed through specific **Routes** configured on the product. For a dropshipped product, you must activate the "Buy" route and ensure the "Dropship" route (or a custom route configured for dropshipping) is selected. This tells Odoo that demand for this product should trigger a purchase order, with the final delivery address being the customer's, rather than your own warehouse. **Example:**

    1. Go to Inventory > Products > Gadget X.
    2. Ensure Product Type is Storable Product.
    3. Go to the Inventory tab. Under "Routes," ensure Buy is checked, and specifically, the Dropship route is also checked (you might need

to enable the Dropshipping feature in Inventory > Configuration > Settings).

4. Ensure Dropship Corp is set as a vendor on Gadget X's Purchase tab. Now, when a sales order is confirmed for Gadget X, Odoo automatically generates a Purchase Order for Dropship Corp, with the Delivery Address on the PO being your customer's address.

3. **Question:** Your finance department requires that certain expenses related to purchasing (e.g., freight, customs duties) be added to the cost of the imported goods to ensure accurate inventory valuation. How can these "Landed Costs" be recorded and correctly allocated to purchased products within Odoo?

   1. A) By manually adjusting the cost price of products after each import.
   2. B) Creating separate vendor bills for each landed cost component and linking them to the Purchase Order.
   3. C) Utilizing the "Landed Costs" feature in the Inventory module, linking them to receipts.
   4. D) Adding freight and duties as separate lines on the original Purchase Order.

4. **Answer: C**

   Explanation & Example: Odoo's **"Landed Costs"** feature (found in the Inventory module settings) is specifically designed to add additional costs (like shipping, customs, insurance) to the cost of your products *after* they have been received. These costs are then proportionally allocated to the items on the linked receipts, ensuring an accurate valuation of your inventory. **Example:**

   1. Enable "Landed Costs" in Inventory > Configuration > Settings.
   2. Receive a shipment of "Imported Goods" via a Purchase Order.
   3. Receive a separate invoice for freight ($200) and customs ($100).
   4. Go to Inventory > Operations > Landed Costs.
   5. Click Create. Select the Transfer (receipt) you want to apply costs to.
   6. Add Cost Lines for "Freight" and "Customs," entering their respective amounts.
   7. Click Compute and Validate. Odoo will then add these costs to the cost price of the "Imported Goods" in your stock, impacting your inventory valuation and COGS.

5. **Scenario:** A batch of 50 "Electronic Chips" was received from Vendor Z via Purchase Order PO001. Upon quality inspection, 5 units are found to be defective and need to be returned to Vendor Z. How would you initiate this return process from the Purchase module in Odoo?

   1. A) Create a new Purchase Order for negative quantity to Vendor Z.
   2. B) Create a manual inventory adjustment to remove the 5 units.
   3. C) From the original Receipt linked to PO001, use the "Return" button to generate a return picking.
   4. D) Create a new "Scrap" order for the 5 units.
6. **Answer: C**

   Explanation & Example: The most proper way to return goods to a vendor that were originally received via a purchase order is to use the **"Return" functionality directly from the original Receipt (Inventory Transfer)**. This ensures proper linkage to the inbound move, accurate stock adjustments, and facilitates any associated vendor credit notes. **Example:**

   1. Go to Purchase > Orders > Purchase Orders. Open PO001.
   2. Click the Receipts smart button.
   3. Open the Done receipt document for the 50 "Electronic Chips."
   4. Click the Return button.
   5. In the wizard, specify the Quantity to Return (5 units of "Electronic Chips") and the Return Location (e.g., your default stock location).
   6. Click Return. This generates a new outgoing picking (delivery order for the vendor) for those 5 units, effectively reversing the stock movement.
7. **Question:** Your procurement team is consolidating several small, unrelated demands for common office supplies (e.g., 2 pens for one department, 5 notebooks for another, 1 stapler for a third) into a single purchase order to a preferred vendor, "Office Supplies Inc.," to leverage bulk discounts and reduce shipping costs. Which feature allows them to group these demands into one RFQ?

   1. A) Creating individual "Request for Quotations" for each demand and merging them manually.
   2. B) Setting up a "Blanket Order" for each specific office supply.
   3. C) Using "Purchase Requisitions" or generating RFQs from the "Procurement Planner" that then consolidate automatically or semi-automatically.
   4. D) Only manual aggregation on a single purchase order is possible.
8. **Answer: C**

   Explanation & Example: Odoo facilitates consolidating demands through a few mechanisms, but the most direct way to *group* demands into a single RFQ is often through **"Purchase Requisitions"** (if that module is used/enabled) or by leveraging the results from the **"Procurement Scheduler."** The scheduler (triggered by reordering rules or MTO) can create individual draft RFQs that can then be reviewed and grouped into a single consolidated RFQ/PO before sending. **Example:**

1. Several individual needs arise for office supplies, which might trigger separate reordering rules or manual RFQ creations (initially in draft).
2. Go to Purchase > Orders > Requests for Quotation.
3. Select multiple draft RFQs that are for the same vendor or can be sourced from the same vendor.
4. Odoo (depending on specific configuration/version) might offer an option to "Group RFQs" or you manually create one new RFQ for "Office Supplies Inc." and add all the required products/quantities from the multiple internal demands onto this single RFQ, then send it. This allows for bulk purchasing and simplified logistics.

9. **Scenario:** For all incoming receipts of "Pharmaceutical Product A," your company requires a mandatory quality control step before the product is moved into usable stock. Only after passing QC should the product be available for sales or internal use. How would you configure this multi-step receipt flow in Odoo?

   1. A) Manually perform QC and then manually transfer stock to the main warehouse.
   2. B) Configure a "Multi-Step Route" for receipts (e.g., Vendor -> Input -> Quality Control -> Stock).
   3. C) Use a "Scrap" order if the product fails quality control.
   4. D) Set the product type to "Consumable" to bypass stock management.

10. **Answer: B**

    Explanation & Example: Odoo's **"Multi-Step Routes"** in the Inventory module are perfect for managing complex inbound (and outbound) workflows. You can define a series of internal transfers that products must go through upon receipt. For quality control, a common setup is Vendor Location -> Input Location -> Quality Control Location -> Stock Location. **Example:**

    1. Go to Inventory > Configuration > Settings.
    2. Under the "Warehouse" section, enable Multi-Step Routes and configure your warehouse for 3-Step Receipts (Input -> Quality Control -> Stock).
    3. When a Purchase Order for "Pharmaceutical Product A" is confirmed, and its receipt is validated, the first transfer will move it from the Vendor Location to WH/Input.
    4. A second internal transfer will then be created from WH/Input to WH/Quality Control.

5. Only after the WH/Input -> WH/Quality Control transfer is validated (signifying QC is complete and passed) will a *third* transfer be created from WH/Quality Control to WH/Stock, making the product available. This ensures strict adherence to QC protocols.