

Odoo 18 Enterprise Point of Sale MCQs with Balanced Answers & Real-time Explanations

Module: Point of Sale

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. PoS Session Management & Reconciliation

1. **Scenario:** Your retail store manager is closing the PoS session for the day. Odoo's expected cash total is \$1,250. However, after physically counting the cash drawer, the manager finds only \$1,245. How should the manager record this discrepancy in Odoo during the session closing process, and what will be the accounting implication?
 1. A) Adjust the expected cash to \$1,245, resulting in a perfectly balanced session.
 2. B) Enter the counted amount as \$1,245. Odoo will record a \$5 "Cash Difference" (Loss) to a designated account.
 3. C) The session cannot be closed until the cash matches the expected amount exactly.
 4. D) Manually create a journal entry for \$5 debiting a "Cash Shortage" account.

2. **Answer: B**

Explanation & Example: Odoo's PoS cash control ([Point of Sale > Configuration > Point of Sale > Your PoS > Cash Control](#)) is designed to track discrepancies. When closing a session, the manager enters the *actual counted amount*. If it differs from the system's expected amount, Odoo automatically calculates the difference (shortage or overage) and records it to a predefined "Cash Difference" or "Loss/Gain" account, ensuring the cash journal entry balances. **Example:**

1. At the end of the day, in the PoS interface, the manager clicks [Close Session](#).
2. In the cash reconciliation screen, Odoo shows [Expected: \\$1,250](#).
3. The manager enters [Counted: \\$1,245](#).
4. Upon [Validate & Close](#), Odoo will automatically generate a journal entry that includes a debit to a configured "Cash Shortage" or "Cash Difference (Loss)" account for \$5, ensuring the bank account reflects the actual cash deposited and the books balance.

II. Sales Operations & Customer Interaction

2. **Question:** A customer, "Sarah Johnson," frequently shops at your store and is part of your loyalty program. During a PoS transaction, she purchases items totaling \$100. She wants to redeem a loyalty reward of \$10 off, which requires 100 loyalty points. After the discount, she wants to pay the remaining balance using a Gift Card for \$50 and the rest with cash. How would the PoS system handle this complex transaction in a single flow?

1. A) Process the sale, then manually apply a \$10 discount, then process gift card payment, then cash payment. Loyalty points are handled separately.
2. B) Select Sarah Johnson, apply the loyalty reward, then process the Gift Card payment, then process the cash payment. The system validates loyalty points and updates balances automatically.
3. C) The system can only handle one payment method per transaction, requiring separate transactions for gift card and cash.
4. D) Loyalty rewards and gift cards cannot be combined in a single PoS transaction.

3. **Answer: B**

Explanation & Example: Odoo's PoS is designed for comprehensive customer management, including loyalty programs and multiple payment methods within a single transaction. Once the customer is identified, loyalty points/rewards, gift cards, and various payment types can be applied seamlessly. **Example:**

1. In the PoS interface, click the **Customer** icon and select **Sarah Johnson**.
2. Add products totaling \$100 to the cart.
3. Click the **Loyalty Program** button/tab. If Sarah has enough points (100+), the \$10 reward option will appear. Click to apply it. The total reduces to \$90.
4. Click the **Payment** button.
5. Select **Gift Card** payment method. Enter **\$50**. Odoo will validate the gift card balance.
6. The remaining balance is \$40. Select **Cash** payment method and enter **\$40**.
7. **Validate**. The system records the sale, reduces Sarah's loyalty points by 100, deducts \$50 from the gift card balance, records the \$40 cash payment, and updates inventory/accounting.

4. **Scenario:** A customer wishes to return a "Blue T-shirt" they purchased yesterday via PoS. They do not have the original receipt but can provide the exact date and approximate time of purchase. How can the cashier efficiently process this return in Odoo PoS while ensuring proper inventory and accounting reversal, especially if the original transaction included loyalty points or a discount?

1. A) Process a new negative sale for "Blue T-shirt" and manually adjust loyalty points.
2. B) Initiate a "Return" process from the PoS interface, search for the original order using date/customer info, select the item(s) to return, and process the refund.
3. C) The return cannot be processed without the original receipt barcode.
4. D) Create a credit note in the Accounting module and then manually adjust inventory.

5. **Answer: B**

Explanation & Example: Odoo PoS allows for efficient returns by linking them back to the original sales order, even without the physical receipt barcode, provided enough information is available to find the original order (e.g., date, customer name, approximate time). This ensures that discounts, loyalty points, and payment methods are handled correctly during the reversal. **Example:**

1. In the PoS interface, go to the **Orders** section (often accessible via a specific button or the main menu).
2. Use the **Filters** or **Search** bar to find the original order (e.g., filter by **Date: Yesterday**, then **Customer: [if known]** or scroll through orders from that time).
3. Once the original order for "Blue T-shirt" is found, click on it.
4. Select the **Return** button. Odoo will then create a new PoS order with negative quantities for the selected items.
5. Adjust quantities if only part of the original order is being returned. The system automatically reverses loyalty points earned, applies original discounts, and suggests the refund amount.
6. Process the refund using the desired payment method (e.g., cash, credit back to original payment method). Odoo automatically handles the inventory return and generates the accounting entries (credit note).

III. Product & Pricing Management

4. **Question:** Your PoS system is configured to display product stock. You have "Product A" with an "On Hand" quantity of 5 units. A customer attempts to purchase 10 units. Your PoS settings allow "negative stock." What will be the immediate impact on "Product A"'s inventory and the resulting accounting entry if the sale is completed?

1. A) The sale will be blocked, as stock is insufficient.
2. B) The "On Hand" quantity will become -5. The Cost of Goods Sold (COGS) will be recorded for 10 units at the standard cost, potentially leading to incorrect profit margins.
3. C) The "On Hand" quantity will remain 5, and the customer will be automatically backordered for 5 units.
4. D) The sale will complete, but no COGS will be recorded for the 5 units sold in negative stock.

5. **Answer: B**

Explanation & Example: If Odoo's inventory settings allow for "negative stock," the system will permit sales even when physical stock is insufficient. The immediate consequence is that the **On Hand** quantity goes below zero. From an accounting perspective, Odoo will still try to record the **Cost of Goods Sold (COGS)** for the full quantity sold based on the product's valuation method. This can lead to misleading profitability figures until the negative stock is resolved. **Example:**

1. **Product:** Product A, **On Hand:** 5 units. Cost: \$10/unit.
2. PoS setting: **Allow Negative Stock** is enabled.
3. Customer buys 10 units of "Product A" at PoS.

4. Upon **Validate** payment, Odoo records the sale.
5. **Product A's On Hand** quantity in inventory becomes $5 - 10 = -5$.
6. Odoo's accounting entries will debit **COGS** for **\$100** ($10 \text{ units} * \$10/\text{unit}$) and credit **Inventory** for **\$100**, even though only \$50 worth of actual stock left the warehouse. This creates an imbalance that needs to be corrected when the negative stock is resolved through replenishment.

IV. Offline Mode & Hardware Integration

5. **Scenario:** Your retail store's internet connection unexpectedly drops in the middle of a busy sales period. The PoS system is configured for "Offline Mode." A cashier continues to process several cash and card transactions. What happens when the internet connection is restored, and the PoS session is reconnected?
 1. A) All transactions processed offline are lost and must be re-entered manually.
 2. B) Transactions processed offline are immediately synchronized to the central Odoo database, including sales orders, payments, and inventory updates. Card payments processed offline will need manual verification.
 3. C) Only cash transactions are synchronized; card transactions are not recorded.
 4. D) The PoS session will freeze until the connection is restored, preventing any further sales.

6. **Answer: B**

Explanation & Example: Odoo's PoS "Offline Mode" is a robust feature designed for business continuity. When enabled, the PoS terminal stores transactions locally. Once the internet connection is restored, these pending transactions are **automatically synchronized** with the central Odoo database, ensuring all sales, payments, and inventory changes are accurately recorded. For card payments, this usually means the *transaction details* are sent, but the actual payment processing would have occurred via an integrated payment terminal which likely requires a connection itself, or if it was a manual process, it would rely on external reconciliation. The core PoS data synchronization is automatic. **Example:**

1. PoS **WH/Main/PoS** goes offline.
2. Cashier processes 3 cash sales and 2 credit card sales (assuming integrated payment terminal can queue/process offline or local manual payment confirmation is done).
3. Internet restores. The PoS automatically detects the connection.
4. All 5 sales orders are pushed to the main Odoo database. Inventory is updated, payments are registered, and accounting entries are generated as if they happened online. The manager can then review these transactions from the backend **Point of Sale > Orders** menu.

V. Reporting & Analysis

6. **Question:** Your sales manager wants to analyze the average order value and the most popular product categories sold through a specific PoS outlet (**PoS Outlet A**)

during last quarter, considering all payment methods. Which Odoo report would provide this consolidated view?

1. A) "Sales by Customer" report, filtered by PoS outlet.
2. B) "Profit and Loss" report, filtered by "PoS Outlet A."
3. C) "Point of Sale Analysis" report (Pivot Table), with measures for "Total" and grouping by "Product Category."
4. D) "General Ledger" for all sales accounts.

7. **Answer: C**

Explanation & Example: The **"Point of Sale Analysis" report** ([Point of Sale > Reporting > Point of Sale Analysis](#)) is the primary analytical tool for PoS data. It's a pivot table that allows for highly flexible reporting, enabling users to group by various dimensions (PoS outlet, date, product, category, salesperson) and apply different measures (total, average, quantity, profit). **Example:**

1. Go to [Point of Sale > Reporting > Point of Sale Analysis](#).
2. Switch to [Pivot](#) view if not already there.
3. Drag [PoS Outlet](#) to [Filters](#) and select [PoS Outlet A](#).
4. Drag [Order Date](#) to [Filters](#) and select "Last Quarter."
5. Drag [Product Category](#) to [Rows](#).
6. Add [Total](#) (for sum of sales) and [Average Price](#) (for average order value) to [Measures](#). This will generate a clear report showing the total sales and average transaction value for each product category specifically for [PoS Outlet A](#) during the last quarter.

More Advanced Odoo 18 Enterprise Point of Sale (PoS) MCQs

Module: Point of Sale

I. Product Bundles & Pricing Complexity

1. **Scenario:** Your store sells "Gaming PC Bundle" as a single PoS product for \$1,500. This bundle consists of a "CPU Unit" (\$1000 cost), a "Monitor" (\$300 cost), and a "Keyboard & Mouse Combo" (\$200 cost). You have configured this bundle as a BoM (Bill of Materials) for manufacturing, but you want to sell it as a single item in PoS. When a "Gaming PC Bundle" is sold through PoS, what is the accounting and inventory impact of this transaction?
 1. A) The "Gaming PC Bundle" is debited from inventory; its cost is recorded as COGS.
 2. B) The individual components ("CPU Unit," "Monitor," "Keyboard & Mouse Combo") are debited from inventory; their combined cost is recorded as COGS.
 3. C) Only the "Gaming PC Bundle" product's sales revenue is recorded; inventory adjustments are manual.
 4. D) The system will prompt the cashier to manually pick individual components from stock.

2. **Answer: B**

Explanation & Example: When a product in Odoo (like a "Gaming PC Bundle") is configured as a **Kit (BoM type: Kit)**, selling it in PoS (or Sales) will **consume the individual components** from inventory and trigger the COGS calculation based on the *cost of those individual components*, not the kit itself. The kit product itself typically doesn't have an on-hand quantity; its quantity is derived from its components. **Example:**

1. Go to **Manufacturing > Products > Gaming PC Bundle**. Set its **Product Type** to **Storable Product**.
 2. Create a **Bill of Materials** for "Gaming PC Bundle" with **BoM Type** as **Kit**. Add components: "CPU Unit," "Monitor," "Keyboard & Mouse Combo."
 3. In PoS, add "Gaming PC Bundle" to the order.
 4. Upon **Validation** of the PoS order, Odoo will automatically:
 - Reduce the **On Hand** quantity of "CPU Unit," "Monitor," and "Keyboard & Mouse Combo" from your PoS stock location.
 - Generate accounting entries that debit **Cost of Goods Sold** with the sum of the costs of the consumed components (e.g., \$1000 + \$300 + \$200 = \$1500) and credit your **Inventory Valuation** accounts for each component.
3. **Question:** Your store runs a promotion where customers get 10% off their entire order if the total exceeds \$200. Additionally, specific "Premium Brand" products have a 5% discount applied via a pricelist. If a customer buys \$250 worth of items,

including \$100 of "Premium Brand" products, how would Odoo PoS typically apply these discounts if both are active and set up correctly?

1. A) Both discounts stack additively, resulting in 15% off "Premium Brand" items and 10% off others.
2. B) Only the 10% overall order discount applies, as it's a higher percentage.
3. C) Odoo applies discounts based on a configured hierarchy, often applying pricelist discounts first, then promotions, potentially resulting in a combined effect.
4. D) The cashier must manually choose which discount to apply.

4. **Answer: C**

Explanation & Example: Odoo's pricing and discount logic follows a defined hierarchy to prevent unexpected outcomes when multiple rules apply. While the exact stacking behavior can be configured (e.g., exclusive, best price), generally, pricelist rules often apply first, followed by broader promotions or manual discounts. The PoS will calculate the most beneficial or appropriate price based on the setup. In this case, a product-specific pricelist discount usually applies *before* a global order discount. **Example:**

1. **Pricelist:** A pricelist rule for "Premium Brand" products to get 5% off (applied at product level).
2. **PoS Promotion:** A global promotion for 10% off orders over \$200 (applied at order level).
3. Customer order: \$100 "Premium Brand" (pre-discount), \$150 other items. Total \$250.
4. Odoo's calculation:
 - "Premium Brand" items first get their 5% pricelist discount: $\$100 * 5\% = \5 discount.
 - Subtotal becomes: $(\$100 - \$5) + \$150 = \$95 + \$150 = \245 .
 - The overall order (\$245) is still over \$200, so the 10% promotion applies to this new subtotal: $\$245 * 10\% = \24.50 discount.
 - Final Price: $\$245 - \$24.50 = \$220.50$. This scenario demonstrates how different discount types interact, with PoS automatically calculating the final price based on the defined hierarchy.

II. Advanced Customer & Loyalty Features

3. **Scenario:** Your store offers digital "Gift Receipts" for customers buying gifts, which omit prices but include product names and return instructions. After a customer buys several items and requests a gift receipt, what is the best way to generate this specific type of receipt in Odoo PoS?
 1. A) Print the standard receipt and manually cross out prices.
 2. B) Odoo automatically generates a separate "Gift Receipt" option or button upon payment.
 3. C) Configure a specific receipt template that can be selected or triggered based on a PoS order tag.
 4. D) It requires a separate third-party integration or custom development.

4. **Answer: C**

Explanation & Example: Odoo's PoS offers flexibility in receipt printing, often

through configurations on the Point of Sale itself or via custom reporting. While there might not be a direct "Gift Receipt" button out-of-the-box (depending on Odoo version and specific modules), the most common and robust way to achieve this is by configuring **custom receipt templates** that can be triggered conditionally, or by linking them to specific **Order Tags** or **Notes** on the PoS order. **Example:**

1. Go to **Point of Sale > Configuration > Point of Sale > Your PoS**.
 2. Ensure **Order Notes** or **Order Tags** are enabled.
 3. You would typically customize the **receipt.xml** or create a new QWeb report in **Settings > Technical > Reports** to have a **gift_receipt.xml** template.
 4. This template would be designed to hide price fields and potentially include a QR code for easy lookup later for returns.
 5. A custom button in PoS (or a field in **Order Notes**) could be configured to flag the order for a "gift receipt." Upon validation, a custom action could print this specific template instead of the standard one. Alternatively, after the sale, a manager could print it from the backend sales order generated by PoS.
5. **Question:** A cashier attempts to apply an "Employee Discount" of 20% to an entire PoS order for an employee. However, the system prompts for a manager's PIN before applying the discount. What Odoo PoS security feature is likely enforcing this behavior?
1. A) The employee has insufficient loyalty points for the discount.
 2. B) The "Employee Discount" is linked to a "Security Group" that requires an override PIN from a user with higher privileges.
 3. C) The PoS session is in offline mode, blocking special discounts.
 4. D) The discount exceeds the maximum allowed percentage configured for the PoS.
6. **Answer: B**

Explanation & Example: Odoo's PoS allows for fine-grained control over cashier permissions and actions that require managerial oversight. This is typically managed through **User Roles and Access Rights** linked to specific actions or discount types. A "Security Group" can be configured to require a supervisor's PIN for actions that carry financial risk, such as applying large discounts, making significant refunds, or overriding prices. **Example:**

1. Go to **Point of Sale > Configuration > Point of Sale > Your PoS**.
2. Under the **Security** tab, you can enable **Manager Password** or link certain actions to specific **User Groups**.
3. The "Employee Discount" product or action (if configured as a product or a custom button) would have a related security rule. For instance, the **Discount** button or the **Allow Discount** permission for the cashier's user group might be restricted.
4. When the cashier, who does not have manager privileges, tries to apply the "Employee Discount," Odoo identifies that this action requires a higher

permission level and prompts for a PIN from a user with those elevated rights (e.g., the store manager's PIN).

III. Complex Order & Payment Scenarios

5. **Scenario:** A customer wishes to pay for their \$300 PoS order using three different methods: \$100 cash, \$150 with a credit card, and the remaining \$50 with a specific "Store Credit" voucher. How would the cashier accurately process this mixed payment in Odoo PoS?
1. A) Process the cash payment, then the credit card payment, and then create a separate manual journal entry for the store credit.
 2. B) Select each payment method one by one, entering the amount for each, until the total matches the order amount.
 3. C) The PoS system can only handle a maximum of two payment methods per transaction.
 4. D) Create three separate PoS orders, one for each payment method.
6. **Answer: B**
Explanation & Example: Odoo PoS is designed to handle **multiple payment methods** seamlessly within a single transaction. The cashier selects a payment method, enters the amount, and then can proceed to select the next payment method for the remaining balance until the total amount due is covered. **Example:**
1. In the PoS interface, after adding items totalling \$300, click the **Payment** button.
 2. Click **Cash**, enter **100**, and click **Validate**. (Or **Add Payment Line** for Odoo 17+).
 3. Click **Credit Card**, enter **150**, and click **Validate**.
 4. Click **Store Credit**, enter **50**, and click **Validate**.
 5. Once the **Amount Due** becomes **\$0.00**, click **Validate** (or **Process Payment**). Odoo will then close the order, record \$100 to the Cash Journal, \$150 to the Credit Card Journal (or bank clearing), and \$50 to the Store Credit Journal (often a liability account), ensuring full and accurate financial recording for the single sales transaction.
7. **Question:** Your PoS system's payment terminal integration experiences an error after a customer's credit card payment for \$75 is successfully authorized externally, but Odoo does not receive confirmation. The customer leaves with the goods. What is the correct procedure for the cashier to manually record this payment in Odoo to finalize the PoS order and ensure accounting accuracy?
1. A) The cashier should void the order and instruct the customer to re-enter their card details.
 2. B) The cashier should select the "Credit Card" payment method in Odoo, manually enter the \$75 amount, and then manually confirm the payment based on the external authorization.
 3. C) The cashier should mark the order as "Pending" and wait for IT to resolve the integration error.
 4. D) The cashier should record it as a cash payment to close the order.

8. **Answer: B**

Explanation & Example: In scenarios where integrated payment processing fails to send an automatic confirmation to Odoo despite an external authorization, the cashier needs the ability to **manually confirm the payment** within Odoo. This typically involves selecting the payment method (e.g., Credit Card) and manually entering the authorized amount, thereby closing the PoS order and creating the necessary journal entries. **Example:**

1. Customer's card payment for \$75 is approved by the external terminal.
2. Odoo PoS doesn't get the automated confirmation.
3. The cashier clicks **Payment** in PoS.
4. Selects the **Credit Card** payment method.
5. Manually inputs **\$75.00** into the amount field for that payment method.
6. Clicks **Validate**. Odoo will then record the \$75 payment to the credit card journal, update the sales order as paid, and proceed to print the receipt, assuming the cashier verifies the external approval. This allows the business to record the sale and ensures the customer isn't delayed or charged twice.