

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

Module: Inventory

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. Product & Location Management

1. **Scenario:** Your warehouse has multiple storage locations, and you want to ensure that newly received items of "Product X" (a raw material) are always stored in a specific bin location within the "Raw Materials Zone" (WH/Raw/Bin001) if it's empty. If WH/Raw/Bin001 is full, it should go to WH/Raw/Bin002. How would you configure Odoo to automate this storage process?
 1. A) Manually assign the bin location during each receipt.
 2. B) Define a "Putaway Rule" for "Product X" to WH/Raw/Bin001 with a sequence for WH/Raw/Bin002.
 3. C) Set WH/Raw/Bin001 as the default location on "Product X."
 4. D) Implement a "Push Rule" from the Input location to WH/Raw/Bin001.
2. **Answer: B**

Explanation & Example: "Putaway Rules" are the core Odoo feature for automating where products are placed upon receipt or internal transfer within your warehouse. You define rules based on products, product categories, or even packages. These rules dictate the target location and can include sequencing (e.g., try Bin001 first, then Bin002 if Bin001 is full/unavailable). **Example:**

1. Go to **Inventory > Configuration > Putaway Rules**.
2. Click **Create**.
3. Set **Product** to "Product X."
4. Set **When Product Arrives In** to WH/Stock/Input (or whatever your receiving location is).
5. Set **Store To** to WH/Raw/Bin001.

6. Create another Putaway Rule for **Product X** with **When Product Arrives In** set to **WH/Stock/Input** and **Store To** **WH/Raw/Bin002**, ensuring **WH/Raw/Bin001** has a higher (or earlier) sequence/priority if Odoo allows for explicit sequencing, or simply ensuring **Bin001** rule comes first in order. Now, when you validate a receipt for "Product X," Odoo will automatically suggest placing it in **WH/Raw/Bin001**. If that location is full, it will then suggest **WH/Raw/Bin002**.
3. **Question:** Your company sells "Paint Cans" which are purchased in Liters (L) but sold and delivered to customers in Gallons (Gal). You need to ensure accurate stock tracking and automatic unit conversion between these two UoMs. How would you set up "Paint Cans" to handle this? (Assume 1 Gal = 3.785 L)
 1. A) Create two separate products: "Paint Cans (Liters)" and "Paint Cans (Gallons)."
 2. B) Set the "Unit of Measure" on the product to "Liter" and the "Purchase Unit of Measure" to "Gallon."
 3. C) Define "Liter" as the "Unit of Measure" and "Gallon" as the "Purchase Unit of Measure" on the product, ensuring both UoMs belong to the same UoM Category and the conversion factor is correct.
 4. D) Manually calculate and convert quantities for every transaction.
4. **Answer: C**

Explanation & Example: Odoo's **Units of Measure (UoM) system** allows for seamless conversion between different units of the same category (e.g., "Length," "Weight," "Volume"). You define a base UoM for a category (e.g., Liter for Volume) and then other UoMs within that category with their conversion factors. The product then uses a default UoM and can specify a different "Purchase UoM" or "Sales UoM," and Odoo handles conversions automatically. **Example:**

1. Go to **Inventory > Configuration > Units of Measure > Units of Measure**.
2. Ensure "Liter" exists as a UoM in the "Volume" category, with **Reference Unit of Measure for this category** checked.
3. Create "Gallon" as a UoM in the "Volume" category, with a **Ratio** (e.g., 3.785 **times the reference unit of measure**).
4. Open **Product: Paint Cans**.
5. Set **Unit of Measure** to **Liter**.
6. Set **Purchase Unit of Measure** to **Liter**. (Important: The *Purchase UoM* and *Sales UoM* on the product are typically the units you *order in* and *sell in*, respectively. If you buy in liters and sell in gallons, set Sales UoM to Gallon.)

7. Now, if you receive 100 Liters, your stock is 100 Liters. If a Sales Order is created for 10 Gallons, Odoo converts this to 37.85 Liters for picking and stock deduction, but shows 10 Gallons on the Sales Order.

II. Operations (Transfers/Pickings)

3. **Scenario:** Your warehouse has a **WH/Output** location where goods are staged before being loaded onto delivery trucks. When a delivery order is validated from **WH/Stock** to **WH/Output**, you need Odoo to automatically create a subsequent transfer from **WH/Output** to the **Customer Location** as soon as the items arrive at **WH/Output**. How is this continuity configured?
 1. A) By using an "Inventory Adjustment" from **WH/Output** to **Customers**.
 2. B) By setting up a "Push Rule" from **WH/Output** to **Partner Locations/Customers**.
 3. C) By manually creating a new delivery order from **WH/Output** for each shipment.
 4. D) By configuring a "Reordering Rule" on the **WH/Output** location.
4. **Answer: B**

Explanation & Example: "Push Rules" (or "Pushed to" rules within routes) are used to automate transfers *from* one location *to* another, whenever stock arrives at the source location. This is crucial for managing multi-step internal movements like staging or quality control. **Example:**

1. Go to **Inventory > Configuration > Routes**.
2. Find or create the route that involves **WH/Output** to **Customers**.
3. Within that route, define a **Push Rule**:
 - **Operation Type**: Your "Delivery Orders" type.
 - **Source Location**: **WH/Output**.
 - **Destination Location**: **Partner Locations/Customers**.
 - **Supply Method**: "Pull from another Location" (ensuring previous step completes) or direct "Push".
4. Ensure that your main "Delivery" picking type also uses this route. Now, when a picking from **WH/Stock** to **WH/Output** is validated, Odoo sees the incoming stock at **WH/Output** and, because of the Push Rule, automatically creates a new picking from **WH/Output** to **Partner Locations/Customers**, ready for the final loading and delivery.
5. **Question:** A technician reports that 3 units of "Power Supply Unit" are beyond repair and need to be removed from active inventory and disposed of. Which Odoo operation is the most appropriate and traceable way to

record this removal?

1. A) Creating a negative "Inventory Adjustment" for the 3 units.
2. B) Generating a "Scrap" order for the 3 units, linking to a dedicated scrap location.
3. C) Issuing a "Delivery Order" to a virtual "Disposal" partner.
4. D) Deleting the product records from inventory.

6. **Answer: B**

Explanation & Example: The **"Scrap" operation** in Odoo ([Inventory > Operations > Scraps](#)) is specifically designed for removing damaged, obsolete, or otherwise unusable inventory from your active stock. It creates a clear audit trail by moving the items to a virtual "Scrap" location, distinguishing it from regular sales or internal transfers.

Example:

1. Go to [Inventory > Operations > Scraps](#).
2. Click [Create](#).
3. Select [Product: Power Supply Unit](#), [Quantity: 3](#).
4. Select the [Location](#) from which to scrap (e.g., [WH/Stock](#)).
5. The [Scrap Location](#) (usually [Virtual Locations/Scrap](#)) is pre-filled.
6. Click [Validate](#). This action immediately reduces the [On Hand](#) quantity of "Power Supply Unit" from [WH/Stock](#) by 3 and records a traceable scrap move.

7. **Scenario:** During a stock receipt process for 20 units of "Bolt M8," the warehouse worker accidentally scans and validates 25 units. The system allows this. What is the immediate consequence in Odoo's inventory, and how would you correct it without fully reverting the entire receipt?

1. A) The system will automatically generate a warning, and the stock will not be updated.
2. B) The "On Hand" quantity for "Bolt M8" will increase by 25. To correct, a negative "Inventory Adjustment" for 5 units should be created.
3. C) The "On Hand" quantity for "Bolt M8" will increase by 25. To correct, a "Return" to the vendor for 5 units must be created.
4. D) The system will block the validation due to over-receipt.

8. **Answer: B**

Explanation & Example: Odoo typically allows over-receipts unless strict controls are enabled. The immediate consequence is that your **On Hand quantity will be updated to the over-received amount**. The most direct and clean way to correct a minor over-receipt *after* validation, without involving the vendor for a return (unless it's a formal policy), is to

perform a **negative "Inventory Adjustment."** Example:

1. A receipt for 20 "Bolt M8" is confirmed for 25 units. Your **On Hand** quantity for "Bolt M8" increases by 25.
2. To correct, go to **Inventory > Operations > Inventory Adjustments**.
3. Click **Create**.
4. Add a line for **Product: Bolt M8**, specify the **Location** (e.g., **WH/Stock**), and enter the **Counted Quantity** as the *desired* correct **On Hand** quantity (e.g., if you should only have 20 more, and you had 0 before, you enter 20). Alternatively, you can create a new line with a negative **Counted Quantity** for the difference (e.g., -5 units of **Bolt M8** at **WH/Stock** starting from 0).
5. Click **Apply**. This will create a stock move to reduce the stock by 5 units, bringing it to the correct level.

III. Product Tracking & Valuation

6. **Question:** Your company uses "Average Cost" as its inventory valuation method. You purchase 10 units of "Widget A" for \$10 each, then later purchase another 10 units for \$12 each. If you then sell 5 units of "Widget A," what will be the Cost of Goods Sold (COGS) for those 5 units?
 1. A) \$50 (based on FIFO).
 2. B) \$60 (based on LIFO).
 3. C) \$55 (based on the average cost of all available units).
 4. D) \$50 (based on the latest purchase price).
7. **Answer: C**

Explanation & Example: When the "**Average Cost**" valuation method is used, the cost of products sold (COGS) is calculated based on the *average cost of all units available in stock* at the time of the sale, rather than the cost of specific incoming batches (like FIFO or LIFO). **Example:**

1. Initial stock: 0.
2. Receipt 1: 10 units @ \$10/unit. Total value: \$100. Average cost: \$10.
3. Receipt 2: 10 units @ \$12/unit. Total value: \$120.
4. New total units: 20. New total value: \$100 + \$120 = \$220.
5. New average cost: \$220 / 20 units = \$11/unit.
6. When you sell 5 units, the COGS will be **5 units * \$11/unit = \$55**.
This average cost is dynamic and updates with each new purchase.
8. **Scenario:** You have a batch of "Chemical B" (tracked by lots) that is nearing its expiration date. You want to quickly identify all stock of "Chemical B" that expires within the next 30 days to prioritize its use or manage its disposal. Which Odoo Inventory report is most suitable for

this, and what filters/groupings would you use?

1. A) "Inventory Valuation" report, filtered by product and location.
 2. B) "Forecasted Inventory" report, grouped by expiration date.
 3. C) "Lots/Serial Numbers" list view, filtered by product and "Expiration Date" (e.g., "Less than or equal to **today + 30 days**").
 4. D) "Stock Moves" report, filtered by product and inbound date.
9. **Answer: C**

Explanation & Example: The **"Lots/Serial Numbers" list view** (**Inventory > Products > Lots/Serial Numbers**) provides detailed information for each tracked item, including its expiration date (if enabled). This list is highly filterable and sortable, making it ideal for pinpointing specific batches based on their expiry. **Example:**

1. Go to **Inventory > Products > Lots/Serial Numbers**.
2. In the **Search** bar, type "Chemical B" to filter by product.
3. Click **Filters > Add Custom Filter**.
4. Select **Expiration Date is less than or equal to** and then input **current date + 30 days** (e.g., **2025-07-15**).
5. The resulting list will show all lots of "Chemical B" that expire within the next 30 days, along with their current stock and location, allowing for immediate action.

IV. Replenishment & Automation

8. **Scenario:** Your company manufactures "Customized Boards." When a sales order is confirmed for a "Customized Board," it triggers a Manufacturing Order. For some components (e.g., "Resistor A"), you always keep a minimum stock. For others (e.g., "Rare Earth Magnet"), you only purchase them when needed for a specific manufacturing order. How do you configure these components' procurement strategies?
1. A) Set "Resistor A" to "Buy" route with a Reordering Rule, and "Rare Earth Magnet" to "Buy" route with "Make-to-Order."
 2. B) Set both components to "Buy" route with Reordering Rules, but adjust minimums.
 3. C) Set both components to "Buy" route with "Make-to-Order."
 4. D) Manually create purchase orders for "Resistor A" and "Rare Earth Magnet."
9. **Answer: A**

Explanation & Example: This scenario combines two common replenishment strategies in Odoo:

1. **"Resistor A" (Minimum Stock):** This is handled by the **"Buy" route combined with a "Reordering Rule" (Min/Max)**. Odoo ensures that stock levels stay within the defined range, triggering purchases when they drop too low.
2. **"Rare Earth Magnet" (Purchase Only When Needed):** This is handled by the **"Buy" route combined with "Make-to-Order"**. A purchase is generated only when there is a direct demand (e.g., from a Manufacturing Order linked to a Sales Order). **Example:**
3. **Product: Resistor A:**
 - **Inventory tab:** Check **Buy** route.
 - **Reordering Rules:** Create a rule with a **Min Quantity** (e.g., 500) and **Max Quantity** (e.g., 1000).
4. **Product: Rare Earth Magnet:**
 - **Inventory tab:** Check **Buy** route.
 - Ensure the **Buy** route's rule is configured to allow **Make to Order** (this is typically the default for a basic "Buy" route unless modified). No reordering rule is needed here. When a Manufacturing Order for "Customized Board" is created, Odoo will check the availability of both components. If "Resistor A" is below its min, a purchase will be suggested by the scheduler. For "Rare Earth Magnet," a specific RFQ will be generated immediately for the exact quantity needed for that MO.
10. **Scenario:** Your warehouse has received a damaged shipment from a vendor. You've processed the return to vendor. Now you want to prevent these defective products from accidentally being sold or used, even if they are still physically present while awaiting pick-up for return. Which location type or status change would be most appropriate for the holding area of these damaged goods?
 1. A) **WH/Stock** (main stock location).
 2. B) **Virtual Locations/Scrap**.
 3. C) A dedicated internal "Quality Control / Damaged Goods" location with its own rules.
 4. D) **Partner Locations/Vendors**.
11. **Answer: C**

Explanation & Example: While **Virtual Locations/Scrap** (Option B) is for permanently removing items, for temporarily holding damaged goods that might still be returned or require further action (and thus not truly "scrapped"), creating a **dedicated internal location** (e.g., **WH/Damaged**, **WH/QC-Hold**) is best practice. This location can be configured to prevent outgoing moves (or require specific permissions), effectively segregating the stock from usable inventory. **Example:**

1. Create a new **Location** in **Inventory > Configuration > Locations** named **WH/Damaged Goods**.
2. When the damaged goods are returned from **WH/Stock** via a return picking, or during initial receipt, instead of moving them to **WH/Stock**, move them to **WH/Damaged Goods**.
3. This location can be set up as **Is a scrap location** or **Is a production input/output location** or just a regular internal location but with stricter access rights or no putaway/removal rules, making it a "dead end" for sales/manufacturing. This keeps the damaged items out of regular stock but trackable.

V. Barcode & Physical Inventory

10. **Question:** Your warehouse team uses barcode scanners for all inventory operations. When validating a delivery order, a picker scans a product's barcode and then its associated lot number. What is the immediate function of scanning the lot number in this context?

1. A) To update the product's cost price for that lot.
2. B) To ensure the correct lot is picked according to the removal strategy and to record its outbound movement for traceability.
3. C) To automatically generate a new reordering rule for that lot.
4. D) To mark the product as expired.

11. **Answer: B**

Explanation & Example: Scanning a lot number during a picking operation serves two critical functions: it ensures that the picker selects the **correct lot** (especially when removal strategies like FEFO or FIFO are active), and it precisely **records the movement of that specific lot** from its location to the customer. This maintains complete traceability for all lot-tracked products. **Example:**

1. A delivery order for "Medicine X" (Lot A, Expiry 2025-12-31) and "Medicine X" (Lot B, Expiry 2026-06-30) is presented to the picker, and the removal strategy is FEFO.
 2. The picker scans the barcode for "Medicine X."
 3. Then, prompted by Odoo, the picker scans the barcode for **Lot A**.
 4. This action confirms that Lot A is being picked, deducting the quantity from Lot A's stock and updating its movement history. If the picker accidentally scans Lot B, Odoo will typically alert them if Lot A was the suggested lot.
12. **Scenario:** Your company performs annual physical inventory counts. To minimize disruption, you want to freeze inventory movements for a specific warehouse (**WH**) during the counting period, allowing new sales orders to be created but preventing new receipts or deliveries from being

processed until the count is finalized. How would you best manage this in Odoo?

1. A) Manually instruct all warehouse staff to stop operations for WH.
2. B) Create a new "Inventory Adjustment" record for the entire WH and keep it in a draft state during the count.
3. C) Temporarily set the **Allowed Operations** on the WH warehouse record to block incoming/outgoing operations.
4. D) Mark all products in WH as "inactive" during the count.

13. **Answer: B**

Explanation & Example: When performing a full physical inventory count, creating an **"Inventory Adjustment" record for the entire warehouse (or selected locations) and keeping it in a "Draft" or "In Progress" state** effectively freezes inventory. Odoo will generally block further stock moves for products included in an active inventory adjustment until that adjustment is validated or canceled. This ensures data integrity during the counting process. **Example:**

1. Go to **Inventory > Operations > Inventory Adjustments**.
2. Click **Create**.
3. Enter a **Reference** (e.g., "Annual Inventory Count 2025").
4. Select **Location** as **WH/Stock** (or the top-level WH warehouse).
5. You can then select specific products or click **Start Inventory** to load all products in that location.
6. As long as this inventory adjustment record remains in the "In Progress" (or draft, depending on version/config) state, Odoo will typically prevent new receipts or deliveries for the products included in that inventory count, prompting users that "an inventory adjustment is in progress for this product/location."

Some Additional Odoo 18 Enterprise Inventory MCQs

Module: Inventory

1. **Scenario:** Your company operates two warehouses, **WH/North** and **WH/South**. Products are always received at **WH/North** (the main receiving hub), but "Product Z" is frequently transferred to **WH/South** for regional distribution. You have a **Push Rule** configured from **WH/North/Stock** to **WH/South/Output** for "Product Z". A sales order from a customer serviced by **WH/South** confirms. How will Odoo attempt to fulfill this order, considering the existing **Push Rule** and a general **Pull Rule** for sales?

1. A) It will immediately create a delivery order from WH/North/Stock to the customer.
2. B) It will first create an internal transfer from WH/North/Stock to WH/South/Input due to the Push Rule, and then a delivery from WH/South/Stock to the customer.
3. C) It will create a direct delivery order from WH/South/Stock to the customer if stock is available there, otherwise, it will create a transfer from WH/North to WH/South only if WH/South stock is insufficient.
4. D) The system will block the sales order until Product Z is manually moved to WH/South.

2. **Answer: C**

Explanation & Example: Odoo's routing engine prioritizes fulfilling demand from the most direct or available source, and it's driven by a "pull" (demand-driven) mechanism. While a Push Rule dictates what happens *when stock arrives* at a location, a sales order generates a *pull* for stock at its intended delivery warehouse (WH/South).

1. Odoo first checks WH/South/Stock for Product Z (due to the delivery rule for sales orders from WH/South).
 2. If Product Z is available in WH/South/Stock, the delivery order is created from there.
 3. If WH/South/Stock is insufficient, the system then looks upstream via its defined routes. The Push Rule from WH/North to WH/South now acts as a *supply path*. Odoo identifies that WH/North can supply WH/South, and thus creates the necessary internal transfer from WH/North to WH/South to satisfy the WH/South sales order. This ensures efficient stock allocation, avoiding unnecessary transfers if stock is already available at the destination.
3. **Question:** You have "Custom Designed T-Shirts" that come in different sizes (S, M, L, XL) and colors (Red, Blue, Green). You manage these as product variants in Odoo. A customer orders 5 "Custom Designed T-Shirts - Blue, Large". How would the warehouse team ensure they pick the exact correct variant using barcode scanning, assuming each variant has a unique barcode?
1. A) Scan the generic "Custom Designed T-Shirts" product barcode and then manually select "Blue, Large" from a list on the scanner.
 2. B) Scan the unique barcode for "Custom Designed T-Shirts - Blue, Large" which directly identifies the variant.
 3. C) Scan the product barcode, then the color barcode, then the size barcode sequentially.
 4. D) Manually enter the product name and then the color and size attributes.

4. **Answer: B**

Explanation & Example: For products with variants, Odoo generates a **unique barcode for each specific variant combination** by default (if barcode is enabled for products). This allows for highly accurate picking and receiving. Scanning the variant's unique barcode directly identifies both the product and its specific attributes (size, color, etc.), eliminating manual selection and reducing errors. **Example:**

1. In **Product: Custom Designed T-Shirts**, you have variants generated for **Color: Red, Blue, Green** and **Size: S, M, L, XL**.
2. Odoo automatically assigns a unique barcode to **Custom Designed T-Shirts - Blue, Large**.
3. When a delivery order requires this specific variant, the warehouse picker simply scans the barcode label on the physical "Blue, Large" T-shirt.
4. Odoo's barcode app or scanning interface immediately recognizes the exact variant and updates the picking line, confirming the correct item is being moved.
5. **Scenario:** Your company imports "Electronics Components" that are expensive and subject to strict quality control upon arrival. You've configured a multi-step receipt route: **Vendor -> Input -> Quality Control -> Stock**. If a shipment of 100 "Electronics Components" arrives, and 10 units fail the QC inspection, how would Odoo most effectively handle the stock movement for the failed units without completely blocking the remaining 90 good units from moving to stock?
 1. A) All 100 units are moved to stock, and then a scrap order for 10 units is created.
 2. B) During the **Input -> Quality Control** transfer, adjust the "Done" quantity to 90. Then, create a separate move for the 10 failed units from **Input** to a "Quarantine" or "Scrap" location.
 3. C) The entire 100-unit transfer is put on hold until all units pass QC.
 4. D) Only 90 units are received from the vendor, and a backorder is created for the 10 units.

6. **Answer: B**

Explanation & Example: When using multi-step routes, partial validations and segregated movements are key. If 10 units fail QC:

1. The initial receipt **Vendor -> Input** will likely be for 100 units.
2. The next step, **Input -> Quality Control**, is where the check happens. Here, the operator validates only the **90 good units** on that specific transfer line. This allows the 90 good units to proceed through the **QC -> Stock** step.

3. For the **10 failed units**, you initiate a *separate stock move* directly from **Input** (or **Quality Control** location if they temporarily went there) to a dedicated "Quarantine" or **Virtual Locations/Scrap** location. This keeps the good units flowing and isolates the bad ones for further action (return to vendor, actual scrap, rework).

Example:

4. An **IN/001** (Receipt) for 100 units of "Electronics Components" is validated. Stock is now at **WH/Input**.
 5. A **QC/001** (Internal Transfer) from **WH/Input** to **WH/Quality Control** is created.
 6. During QC, the operator marks 90 units as **Done** for **QC/001**.
 7. They then go to **Inventory > Operations > Transfers**, **Create** a new **Internal Transfer** from **WH/Input** (or **WH/Quality Control**) to **WH/Damaged Goods** (a custom location) for the 10 failed units. This allows the 90 units to continue to stock while managing the 10 units separately.
7. **Question:** Your Odoo system uses "FIFO" (First In, First Out) for inventory valuation. A product, "Premium Coffee Beans," has the following incoming stock:
1. Batch 1: 50 kg received on May 1st at \$15/kg
 2. Batch 2: 70 kg received on May 15th at \$18/kg You then process a sales delivery for 80 kg of "Premium Coffee Beans" on June 1st. What will be the total Cost of Goods Sold (COGS) recorded for this sale?
 3. A) \$1200 (80 kg * \$15/kg)
 4. B) \$1360 (80 kg * \$17/kg, average cost)
 5. C) \$1290 (50 kg * \$15/kg + 30 kg * \$18/kg)
 6. D) \$1440 (80 kg * \$18/kg)
8. **Answer: C**

Explanation & Example: FIFO (First In, First Out) valuation means that the goods purchased or produced earliest are assumed to be sold first. Therefore, the Cost of Goods Sold (COGS) is calculated using the cost of the oldest inventory until that batch is depleted, then moving to the next oldest. **Example:**

1. **First 50 kg:** Come from Batch 1 at \$15/kg. Cost = $50 * \$15 = \750 .
2. **Remaining 30 kg (80 - 50):** Come from Batch 2 at \$18/kg. Cost = $30 * \$18 = \540 .
3. **Total COGS:** $\$750 + \$540 = \$1290$. This method accurately reflects the flow of goods for businesses where older inventory is typically sold first.

9. **Scenario:** Your company regularly ships products in standardized cartons. A delivery order is created for 120 units of "Product X." You know that "Product X" is packed 10 units per carton. You want the warehouse team to pick full cartons efficiently. How would you configure the product or picking process to guide them to pick cartons rather than individual units, and track the cartons?

1. A) Set the "Unit of Measure" of "Product X" to "Carton."
2. B) Define "Packaging" for "Product X" with 10 units per carton, and ensure the picking type utilizes package tracking.
3. C) Manually instruct the pickers to always pick full cartons.
4. D) Create a "Reordering Rule" for cartons of "Product X."

10. **Answer: B**

Explanation & Example: Odoo's **"Packaging"** feature ([Inventory > Configuration > Packaging](#)) is designed precisely for this. You define different package types for a product (e.g., "Box of 10," "Pallet of 100"). When configuring the picking type or the delivery order itself, Odoo can be set to prioritize picking in full packages, guiding the picker and allowing for accurate tracking of packages. **Example:**

1. Open [Product: Product X](#).
2. Go to the [Inventory](#) tab and click [Packages](#).
3. Click [Create](#). Set [Package](#) as "Carton." Set [Contained Quantity](#) to 10.
4. Ensure that your Delivery Picking Type ([Inventory > Configuration > Picking Types](#)) has "Use existing packages" and/or "Put in pack" configured if you also want to generate new packs. When the delivery order for 120 units of "Product X" is created, Odoo will calculate that 12 cartons are needed. In the "Detailed Operations" on the picking, it will often suggest picking 12 units of "Carton" for "Product X," making the process very clear for the warehouse team and ensuring full cartons are moved.

11. **Scenario:** Your company performs cycle counting for different product categories monthly. For "High-Value Electronics," a full count is done every month. For "Low-Cost Consumables," a count is done quarterly. How would you set up Odoo to automate the generation of these distinct cycle count sheets on a recurring basis?

1. A) Manually create a new "Inventory Adjustment" for each category every month/quarter.
2. B) Configure "Inventory Adjustment" records and use Odoo's automated "Scheduled Actions" to trigger them.
3. C) Utilize the "Inventory Counts" feature, specifying recurrence and filtering by product category.

4. D) Implement a custom script to generate counting sheets.

12. **Answer: C**

Explanation & Example: Odoo's "**Inventory Counts**" (or "Physical Inventory") feature ([Inventory > Operations > Inventory Adjustments > Create > Start Inventory](#) and then filtering options) combined with its scheduling capabilities allows for recurring cycle counts. You can create an inventory adjustment template, filter it by product category or location, and then use Odoo's built-in scheduled actions to generate these counts automatically at defined intervals. **Example:**

1. Go to [Inventory > Operations > Inventory Adjustments](#).
2. Click [Create](#). Give it a reference like "Monthly Cycle Count - Electronics".
3. Select the [Location](#) (e.g., [WH/Stock](#)).
4. Click [Add a line](#) or [Start Inventory](#). Then use [Filters](#) to include [Product Category is High-Value Electronics](#).
5. Save this inventory adjustment (it can be in a draft state).
6. Go to [Settings > Technical > Automation > Scheduled Actions](#).
7. Create a new scheduled action that runs monthly, using the "Create New Inventory Adjustment" action (or similar model/method) and specifying the template or filter criteria for "High-Value Electronics."
8. Repeat for "Low-Cost Consumables" but set the schedule to quarterly. This automates the creation of the count sheets, ready for your team to perform the physical count and apply the adjustments.