



Total Marks: - 30

Total Time:- 6 Hours

- 1) Create one new master table Product Group.
 - a) Product Group table should have two mandatory fields “**Name**” and “**Short Code**”. The field ‘short code’ should be unique and needs to have a prefix “**pg-**”. ‘short code’ will be like - “pg-002”. **(2 Mark)**
 - b) Product Group table should have one many2one field with product group table itself named “**Parent Group**”. **(0.5 Mark)**
 - c) Design **Tree, Form** and **Search** view for product group. **(2 Mark)**
 - d) Add a menu under **Inventory / Configuration / Products**. **(0.5 Mark)**
 - e) Create many2one field “**Group**” in Product table with **product group** and put the field under **product category** in product form view. **(1 Mark)**
- 2) Filter products in Purchase Order line based on Product Group.
 - a) Add a many2one field named “**Product Group**” in Purchase Order with Product Group and placed it under the field **Order Deadline** in purchase order form. **(1 Mark)**
 - b) Add a Boolean field named “**Active Group Filter**” and placed it before ‘**product group**’. If this field enabled then ‘product group’ will be **mandatory** and visible otherwise it will be invisible. **(3 Mark)**
 - c) In purchase order line products will be filtered depending on selected product group. **(2 Mark)**
- 3) Data Validation and Formation of Fields.
 - a) In Customer Table, when we enter an email, it will accept only email from “**gmail.com**” domain. The email should start with and alpha character and and should not contains any special character or spaces. **(2 Mark)**
 - b) If anyone enters Capital letters, on save make it small letters. **(1 Mark)**
- 4) Purchase Order Filter from wizard.
 - a) Add a menu under **Purchase / Orders** named “**Filter Purchase Orders**”. **(1 Mark)**
 - b) On click this menu, an wizard will open. On that wizard, one many2many field will be there linked with customers. **(2 Mark)**
 - c) Wizard will contain a button “**view purchase orders**”, on click that button, purchase order list from those selected customers will open. **(2 Mark)**
- 5) Product Rating
 - a) Add a selection field on Products named “**Product Rating**”. Options as 1 2, 3, 4 5 stars by default it will have 1 start. **(1 Mark)**
 - b) Add Filter on product named “**Rating**” on group by control panel view. **(1 Mark)**

- 6) Qweb Report
- Inherit Purchase Order Report and product group after order date. **(1 Mark)**
 - On Purchase Order Line, Add that product rating field after description. **(1 Mark)**
- 7) Customer / Vendor Partner Selection
- In purchase order form, on vendor field listing should contain only companies. **(1 Mark)**
 - In Sale order form, on customer field listing should contain only companies. **(1 Mark)**
- 8) Security & Access Rights
- The product Group master we have created - **(2 Mark)**
 - Only **Inventory Administrator** group have all access.
 - Inventory User** group will have all access except delete.
 - Purchase User** group have read access only.
 - All access right applied should be on security file.
- 9) User of Library
- We have a dictionary object stored in a variable name “custom_values”. Now have to search if that dictionary have any key with “**default_**” prefix and remove that prefix. **(1 Mark)**

Sample Input:

```
custom_values = { "name": "Pinto", "default_code": "p001", "default_value": 100 }
```

Sample Output:

```
custom_values = { "name": "Pinto", "code": "p001", "value": 100 }
```

- Arrange the element in ascending order and remove duplicate numbers **(1 Mark)**

Sample Input:

```
[20, 39, 34, 20, 24, 20, 10, 11]
```

Sample Output:

```
[20, 39, 34, 24, 10, 11]
```

- We need to sort the dictionary descending order **(1 Mark)**

Sample Input:

```
data = { 2 : "AA", 1 : "BB", 4 : "CC" }
```

Sample Output:

```
data = { 4 : "CC", 2 : "AA", 1 : "BB" }
```