

# Microsoft Dynamics Qualification Exam

## Contents

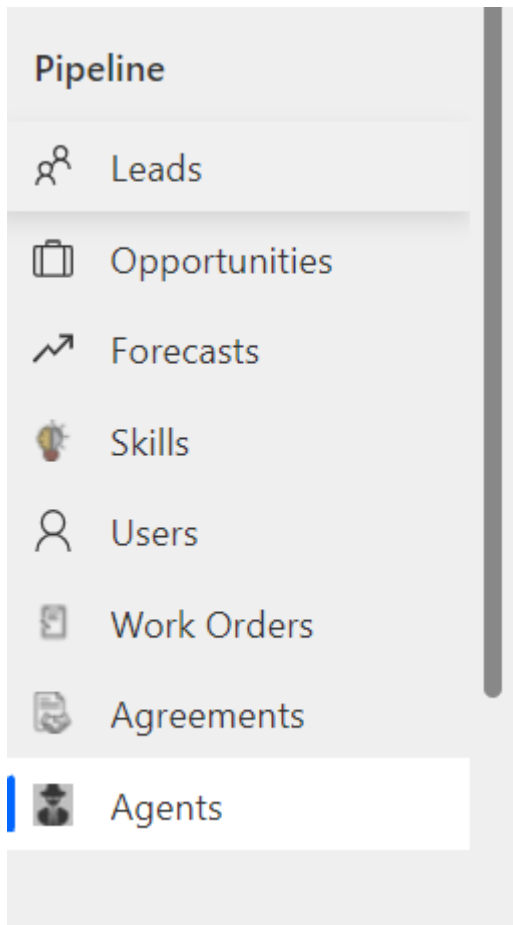
- ❖ Business Scenario 1
- ❖ Business Scenario 2
- ❖ Business Scenario 3
- ❖ Business Scenario 4
- ❖ Theory question

Worked by:Emirion Rystemi

## Introduction

In the Sales Trials, my work is divided into 4 separate solutions, each corresponding to a different business scenario. All customizations are implemented using Visual Studio, utilizing C#, JavaScript, and CRM low-code customizations. Tools such as Xrm Toolbox and Plugin Registration are used development processes.

At the beginning, I created the prefix '**test\_**' and created a project in Visual Studio. I then added all entities and updated their icons in the sitemap at the Pipeline.



# Business Scenario 1

I started by creating a solution for Business Scenario 1. Within this solution, I added a new entity called 'Skill,' defined five skills, and created a 'Proficiency' option set along with the main form. I then established a Many-to-Many relationship between the 'User' and 'Skill' entities to reflect the scenario where users can have multiple skills, and skills can belong to multiple users. Finally, I added a 'Skills' section to the User form, displaying related records for efficient management.

ER

emirion rystemi

- Saved

User · User

Summary

Details

Administration

Related

Skills

C#

4/3/2024 3:41 PM

1 - 1 of 1

Page 1

Country/Region

Other Street 1

Other Street 2

Other Street 3

Other City

Other State/Provin

Other ZIP/Postal C

The main form of the Skills Entity includes a section displaying users with that skill, providing visibility into skill-user relationships.

←

☰

📄

💾 Save

💾 Save & Close

➕ New

🗑️ Deactivate

🗑️ Delete

🔄 Refresh

🔍 Check Access

👤 Assign

⋮

🔗 Share

C# - Saved

Skills

General

Related ▾

Name

\*

C#

Proficiency

2

Users

📋 Add Existing User

🔄 Refresh

⋮

⋮

○

Full Name ↑ ▾

Site ▾

Business Unit ▾

Title ▾

Position ▾

Main Phone ▾

emirion rystemi

org047a54cf

0698672674

The users selects a skill from the dropdown, then filter the list of employees to display only those who possess the selected skill.

joni - Unsav  
Opportunity · Sales Insights

Lead to Opportunity Sal...  
Active for 4 days

<

Qualify

Contractir

Summary

Relationship analytics


Quotes

Products

Related

More info

Skill

 C#

User

---

Current situation

---

joni - Saved  
Opportunity · Sales Insights

Lead to Opportunity Sal...  
Active for 4 days

<

Qualify

Contr

Summary

Relationship analytics

Quotes

Products

Related

More info


Users

ER

emirion rystemi

Advanced lookup

Look for User

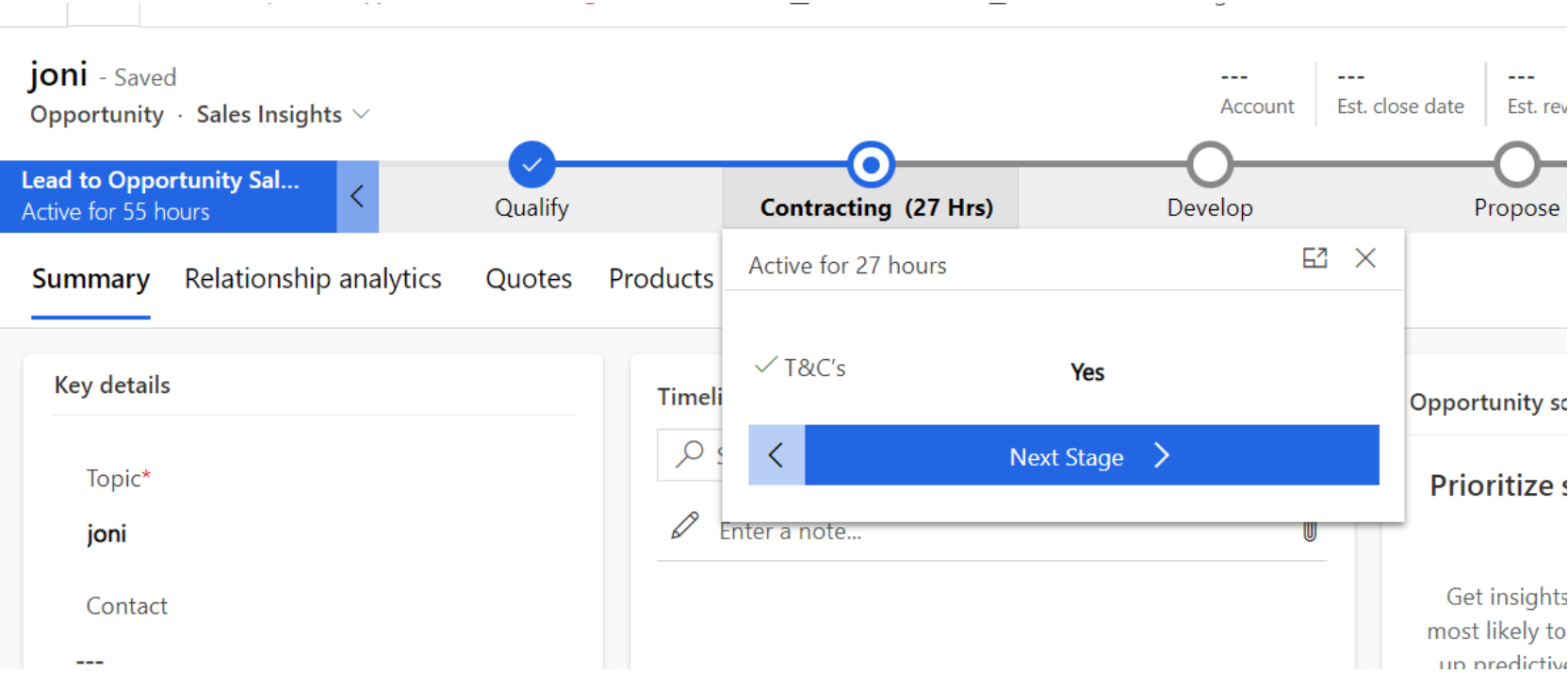


Current situation

---

# Business Scenario 2

After creating a solution for Business Scenario 2, I began by defining an 'Agreement' entity with all necessary columns for the Exam requirements. Upon creating the main form for the Agreement entity, I updated the business process flow by adding a new stage named 'Contracting' and introduced a 'T&C' column with a binary choice (Yes or No). Customizations were implemented to automatically switch T&C to 'Yes' if specific conditions were met, such as having an Onboarding agreement linked to the account associated with the current opportunity, and containing values in both the Agreement start date and end date fields.



I have customized this stage with a Plugin who triggers T&C field with this method:

```
1 reference
71 private void UpdateOpportunities(IOrganizationService service, Entity account)
72 {
73     QueryExpression query = new QueryExpression("opportunity");
74     query.ColumnSet = new ColumnSet("parentaccountid", "statecode");
75
76     FilterExpression filter = new FilterExpression(LogicalOperator.And);
77     ConditionExpression stateCondition = new ConditionExpression("statecode", ConditionOperator.Equal, (int)stateCode.Active);
78     ConditionExpression accountCondition = new ConditionExpression("parentaccountid", ConditionOperator.Equal, account.Id);
79     filter.AddCondition(stateCondition);
80     filter.AddCondition(accountCondition);
81
82     query.Criteria = filter;
83
84     EntityCollection opportunities = service.RetrieveMultiple(query);
85
86     foreach (var opportunity in opportunities.Entities)
87     {
88         Entity updatedOpportunity = new Entity(opportunity.LogicalName, opportunity.Id);
89         bool tc = opportunity.GetAttributeValue<bool>("test_tcs");
90
91         updatedOpportunity["test_tcs"] = true;
92         service.Update(updatedOpportunity);
93         Console.WriteLine("Updated Records" + opportunity.Id.ToString());
94     }
95 }
96
97
98
```

This method updates the test\_tcs field of all active opportunities associated with a specific account to “true”. It retrieves opportunities linked to the provided account, checks if they are active and updates “test\_tcs” to “true” for each opportunity found.

For the other point

❖ (Onboarding and NDA agreements can only be created once for any account).

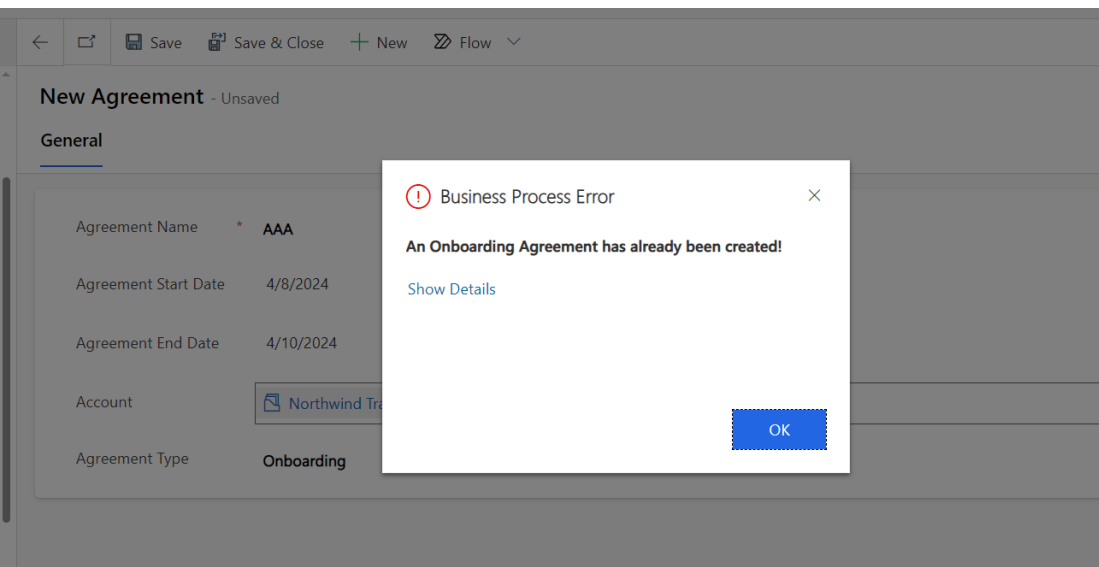
I have created this fetch who contains both of the cases like : Onboarding and NDA and it is called in the function above:

```
1 reference
private void CheckExistingAgreement(IOrganizationService service, Guid accountId, int agreementType)
{
    var fetchXml = @"
        <fetch version='1.0' mapping='logical' no-lock='false' distinct='true'>
            <entity name='test_agreement'>
                <attribute name='test_account' />
                <attribute name='test_agreementtype' />
                <filter type='and'>
                    <condition attribute='statecode' operator='eq' value='{(int)stateCode.Active}' />
                    <condition attribute='test_account' operator='eq' value='{accountId}' />
                    <condition attribute='test_agreementtype' operator='eq' value='{agreementType}' />
                </filter>
            </entity>
        </fetch>";

    EntityCollection agreements = service.RetrieveMultiple(new FetchExpression(fetchXml));

    if (agreements.Entities.Count > 0)
    {
        string agreementTypeName = agreementType == (int)agreementType.Onboarding ? "Onboarding" : (agreementType == (int)agreementType.NDA ? "NDA" : null);
        throw new InvalidPluginExecutionException($"An {agreementTypeName} Agreement has already been created!");
    }
}
```

And shows the following error when you create the same type of agreement with the same account .





# Business Scenario 3

Fist created solution Business Scenario 3

In this solution I added two entities named “Agent” and “Work Order” and created the columns and the main forms.

←

↗

Save

Save & Close


+ New

Flow

New Work Order - Unsaved

General

Assigned Agent

 Joni

Description

---

Work Order Number

23

Duration

---

Start time

4/30/2024

End time

5/30/2024

Scheduled On

Wednesday

Joni - Saved

Agent

General Related

Agent name

\*

Joni

Birthday

4/11/2024

Phone Number

---

Is Scheduled Monday

Yes

Is Scheduled Tuesday

No

Is Scheduled Wednesday

No

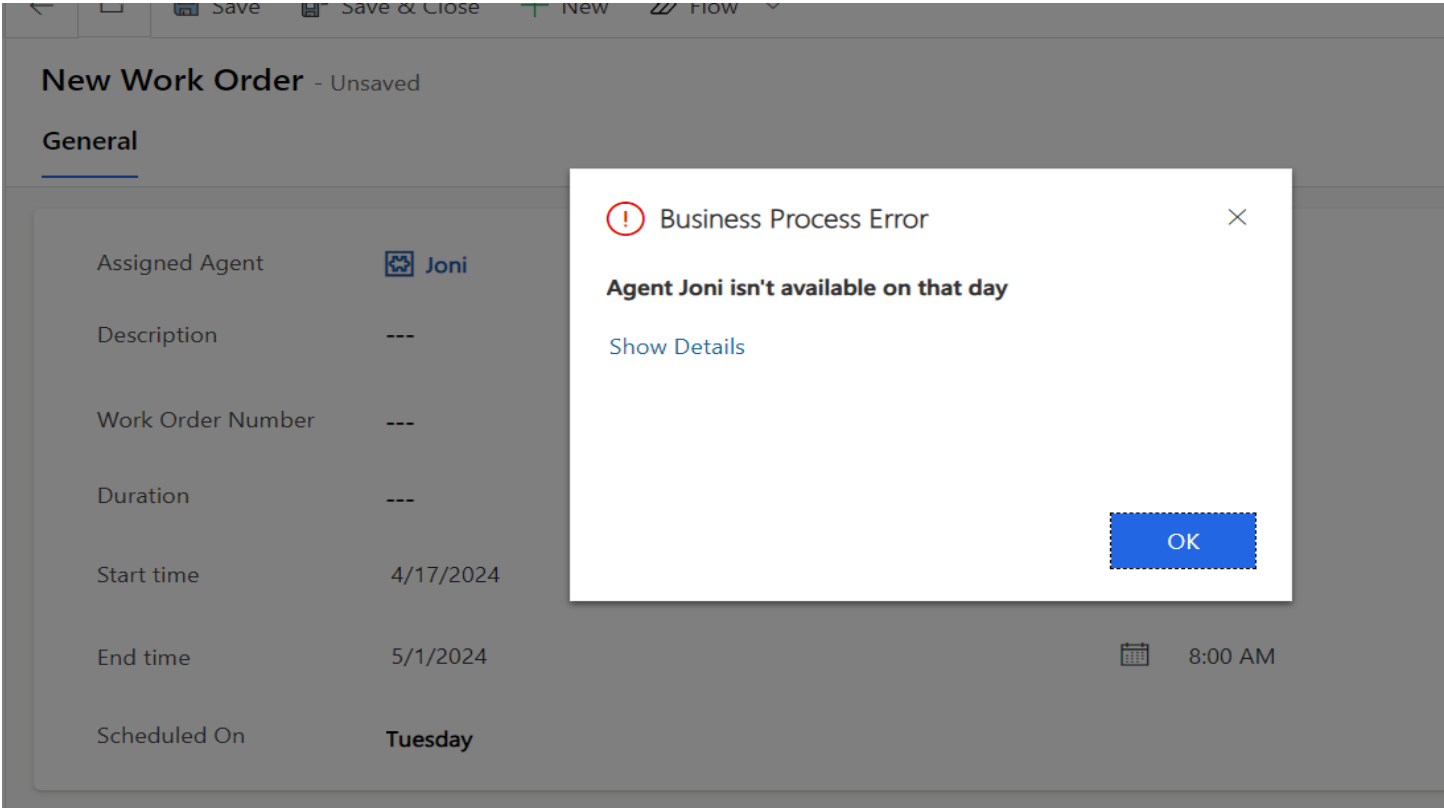
Is Scheduled Thursday

No

Is Scheduled Friday

No

For the issue of preventing the assignment of work orders to agents who aren't scheduled on specific days, I crafted a plugin. This plugin verifies the availability of the assigned agent based on the work order's scheduled day. If the agent isn't available on the specified day, the plugin throws an exception, providing a relevant error message to ensure accurate scheduling.



This is the error when choose Scheduled On Tuesday but the Agent is scheduled on Monday.

# Business Scenario 4

I created a solution for Business Scenario 4. To enforce a specific phone number format, I added a system customization feature, the 'Input Mask Control.' This customization ensures that phone numbers is like the desired format by starting with '+355' and allowing only 9 numbers after.

JO

joni - Saved

Lead · Sales Trial ▾

Lead to Opportunity Sal...  
Active for 3 days

<

Qualify

Contracting (47 Hrs)

Summary

Details

Relationship Analytics

Related ▾

First Name

+

---

Last Name

\*

joni

Job Title

---

Business Phone

+355\_\_\_\_\_

Mobile Phone

---

Email

\*

---

COMPANY

To prevent the creation of leads without an email address, I updated the system by setting the 'Email' field as 'Business Required.'

Quick Create: Lead

Purchase timeframe ---

Lead Source ---

Personal Information

First Name + ---

Last Name \* ---

Job Title ---

Company Name + ---

Contact Details

Email \* ---

Mobile Phone ---

Description ---

Save and Close



A Duplicate Rule was implemented for the 'Business Phone' field. When a user attempts to create a new lead with the same business phone number, the system prompts a notification, indicating the presence of a duplicate record.

Duplicate records found



Merge to an existing record by choosing a record from matched list and clicking merge. To proceed without merging, click Ignore and save. [Learn more](#)

Current record

| ✓ | Name ▾      | Company Name ▾ | Email ▾               | Business Phone ▾ |
|---|-------------|----------------|-----------------------|------------------|
|   | Reuben Kidd | Proseware Inc. | reuben@proseware.c... | 698672674        |

Duplicates found:

Lead (1)

| ✓ | Name ▾ | Email ▾        | Modified On ▾    | Business Phone ▾ |
|---|--------|----------------|------------------|------------------|
|   | joni   | joni@gmail.com | 4/6/2024 9:16 PM | 698672674        |

I have integrated a plugin to automatically append the date after the Topic name, simplifying the process for users.

←

Show Chart

Focused view

+ New

🗑 Delete

Visualize this view

📄 Export to Excel

⋮

| My Open Leads ▾       |                     |  | Edit columns    | Edit filter |
|-----------------------|---------------------|--|-----------------|-------------|
| <input type="radio"/> | Name ▾              | Topic ▾                                    | Status Reason ▾ |             |
|                       | exam exam           | exam 08/04/2024                            | New             |             |
|                       | test test           | new part 08/04/2024                        | New             |             |
|                       | Reuben Kidd         | 10 Airpot Duo Coffee Makers for Prosew...  | New             |             |
| <input type="radio"/> | <u>Jabez Parker</u> | 5 Café Lite Espresso Machines for Prose... | New             |             |

The code executed in the PostOperation stage, used with service.Update.

```
//automatically append the date after the Topic name
string topic = target.GetAttributeValue<string>("subject");
topic += " " + DateTime.UtcNow.ToString("dd/MM/yyyy");

if (stage == (int)Stage.PostOperation)
{
    Entity task = new Entity("task");
    task["subject"] = "Follow Up";
    task["regardingobjectid"] = new EntityReference(target.LogicalName, target.Id);
    service.Create(task);
    target["subject"] = topic;
    service.Update(target);
}
```

When a user creates a lead, the system automatically creates a task named “Follow Up”.

EE

Exam Exam

- Saved

Lead · Sales Trial ▾

Lead to Opportunity Sal...  
Active for less than one mi...

<

Qualify (< 1 Min)

Contracting

Summary

Details

Relationship Analytics

Related ▾

Up next ⓘ

↺

✓ Follow Up

Added by you • Created on 10:26 PM

✓ Task

✓ Mark complete

...

Timeline

+

🔖

🔍

☰

⋮

🔍

Search timeline

## Theory question

**What other types of customizations and extensions can be done in MSDyn365 and when would you choose which? Please prepare an answer.**

- ❖ You can also use Business Process Flows - Business process flows guide users through specific stages and steps to complete specific business processes. They ensure consistency and standardization in workflows across the organization.

Where to use :

Use Workflows for automation tasks that don't require complex logic or external integrations.

- ❖ Another type of customization is Ribbon Buttons – Throw Xrm Toolbox you can add a specific Button located in the ribbon bar that can execute a process etc.

Where to use :

In Business Scenario 1 of the example provided in the exam, we could integrate a button that, when clicked, displays the skills associated with a user.

- ❖ Actions- Actions define specific operations that users can perform, such as creating records, updating data, or executing custom business logic.

Where to use :

Actions in Microsoft Dynamics useful for implementing on multi-step processes that involve multiple entities, consistent execution of complex business logic across the application.