**\*\* AI generated document may be incorrect \*\***

**Technical Specifications for Extended Table (Entity) Objects  
Lead Creation and Conversion to Accounts â€“ Dynamics 365 CE**

This technical specification outlines CRM table (entity) objects and customizations required to implement the described Lead-to-Account conversion process in Dynamics 365 Customer Engagement for a retail company. The content is structured for direct use by technical teams and solution designers.

## 1. Entity (Table) Details

### 1.1 Lead

|  |  |
| --- | --- |
| **Attribute** | **Details** |
| Entity Name | Lead |
| Entity Description | Stores information about potential customers prior to qualification. |
| Primary Key | leadid |
| Auditing | Enabled |
| Enable for Mobile | Yes |

#### Attributes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Display Name** | **Schema Name** | **Type** | **Length** | **Required?** | **Audit Enabled?** |
| First Name | firstname | Single Line of Text | 50 | Yes | Yes |
| Last Name | lastname | Single Line of Text | 50 | Yes | Yes |
| Email Address | emailaddress1 | Email | 100 | Yes | Yes |
| Company Name | companyname | Single Line of Text | 100 | Yes | Yes |
| Lead Source | leadsourcecode | Option Set | - | Yes | Yes |
| Phone | telephone1 | Phone | 20 | No | Yes |
| Status | statuscode | Option Set | - | Yes | Yes |
| Lead Owner | ownerid | Lookup (User/Team) | - | Yes | Yes |

#### Relationships

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Related Entity** | **Relationship Name** | **Description** |
| N:1 | User/Team | lead\_ownerid | Lead assigned to owner |
| 1:N | Activity | lead\_activity\_parties | Lead activities (emails, calls) |
| N:1 | Account (on conversion) | lead\_parentaccountid | Related account after conversion |

### 1.2 Account

|  |  |
| --- | --- |
| **Attribute** | **Details** |
| Entity Name | Account |
| Entity Description | Stores organization/company information. |
| Primary Key | accountid |
| Auditing | Enabled |
| Enable for Mobile | Yes |

### 1.3 Contact

|  |  |
| --- | --- |
| **Attribute** | **Details** |
| Entity Name | Contact |
| Entity Description | Stores individual customer information. |
| Primary Key | contactid |
| Auditing | Enabled |
| Enable for Mobile | Yes |

### 1.4 Opportunity

|  |  |
| --- | --- |
| **Attribute** | **Details** |
| Entity Name | Opportunity |
| Entity Description | Sales opportunity generated from qualified leads. |
| Primary Key | opportunityid |
| Auditing | Enabled |
| Enable for Mobile | Yes |

## 2. Security Roles Matrix

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Role** | **Read** | **Write** | **Create** | **Delete** | **Append** | **Append To** | **Assign** | **Share** |
| Sales Rep | Yes | Yes | Yes | No | Yes | Yes | Yes | No |
| CRM Admin | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes |
| Manager | Yes | No | No | No | Yes | Yes | No | Yes |

## 3. Forms

* **Main Form (Information):** For lead creation, qualification, and conversion (all fields and business process flow).
* **Quick Create Form:** Rapid lead entry (First Name, Last Name, Email, Company, Lead Source).
* **Quick View Form:** For viewing lead details from related Account/Opportunity.

## 4. Views

* **My Open Leads** (default view)
* **All Leads**
* **Leads Awaiting Qualification**
* **Duplicate Leads**
* **Recently Converted Leads**

## 5. Dashboards

* **Lead Pipeline Dashboard:** Charts for leads by status, source, conversion rates, and assignment.
* **Sales Conversion Analytics:** Conversion time, lead source effectiveness, lost vs. converted leads.

## 6. JavaScript Business Logic

**Objective:** To enforce business required field validation and implement dynamic logic on forms.  
**Sample Implementation:**

// On Save: Validate required fields and email/phone format function validateLeadRequiredFields(executionContext) { var formContext = executionContext.getFormContext(); var requiredFields = ["firstname", "lastname", "emailaddress1", "companyname", "leadsourcecode"]; var missingFields = []; requiredFields.forEach(function(field) { var value = formContext.getAttribute(field).getValue(); if (!value || (typeof value === "string" && value.trim() === "")) { missingFields.push(field); } }); // Email format validation var email = formContext.getAttribute("emailaddress1").getValue(); var emailPattern = /^[^@\s]+@[^@\s]+\.[^@\s]+$/; if (email && !emailPattern.test(email)) { formContext.ui.setFormNotification("Invalid email address format.", "ERROR", "email\_error"); executionContext.getEventArgs().preventDefault(); return; } // Phone format validation (example: +1-555-555-5555) var phone = formContext.getAttribute("telephone1").getValue(); var phonePattern = /^\+\d{1,3}-\d{3}-\d{3}-\d{4}$/; if (phone && !phonePattern.test(phone)) { formContext.ui.setFormNotification("Invalid phone format: use +1-555-555-5555.", "ERROR", "phone\_error"); executionContext.getEventArgs().preventDefault(); return; } if (missingFields.length > 0) { var missingLabels = missingFields.join(", "); formContext.ui.setFormNotification("Please fill required fields: " + missingLabels, "ERROR", "required\_fields"); executionContext.getEventArgs().preventDefault(); } }

**Usage:** Attach this JS function to OnSave event for the Lead Main/Quick Create forms.

## 7. Workflow Details

* **Lead Assignment Workflow:** *Custom workflow recommended for round-robin, territory, or score-based assignment.*
* **Lead Conversion Notification:** *Out-of-box workflow to notify sales rep/manager on conversion.*
* **Bulk Lead Conversion:** *Custom workflow for batch processing.*

**Sample Workflow Steps for Assignment:**

1. On lead creation, evaluate assignment rules (territory, round-robin, score).
2. Assign lead to appropriate sales rep.
3. Send notification email.

## 8. Plugin Details

* **Duplicate Detection Plugin** (C#): Checks for duplicate leads using fuzzy matching on company name/email before creation/conversion.
* **Conversion Rollback Plugin** (C#): Ensures rollback if error during conversion; logs exception and notifies user.

**Sample Duplicate Detection Plugin (C#):**

public class LeadDuplicateDetectionPlugin : IPlugin { public void Execute(IServiceProvider serviceProvider) { IPluginExecutionContext context = (IPluginExecutionContext)serviceProvider.GetService(typeof(IPluginExecutionContext)); IOrganizationServiceFactory serviceFactory = (IOrganizationServiceFactory)serviceProvider.GetService(typeof(IOrganizationServiceFactory)); IOrganizationService service = serviceFactory.CreateOrganizationService(context.UserId);

Code

if (context.InputParameters.Contains("Target") && context.InputParameters["Target"] is Entity entity && entity.LogicalName == "lead")

{

var email = entity.GetAttributeValue("emailaddress1");

var company = entity.GetAttributeValue("companyname");

QueryExpression query = new QueryExpression("lead")

{

ColumnSet = new ColumnSet("leadid")

};

query.Criteria.AddCondition("emailaddress1", ConditionOperator.Equal, email);

query.Criteria.AddCondition("companyname", ConditionOperator.Like, "%" + company + "%");

var results = service.RetrieveMultiple(query);

if (results.Entities.Count > 0)

{

throw new InvalidPluginExecutionException("Potential duplicate lead detected.");

}

}

}

}

## 9. Azure Function Details

* **Integration:** For web form/API lead creation, call Azure Function to validate and push leads into CE securely.

**Sample Azure Function (C#):**

[FunctionName("CreateLeadFromWebForm")] public static async Task Run( [HttpTrigger(AuthorizationLevel.Function, "post", Route = null)] HttpRequest req, ILogger log) { string requestBody = await new StreamReader(req.Body).ReadToEndAsync(); var data = JsonConvert.DeserializeObject(requestBody); // Validate data, connect to CE, create lead record // Return result return new OkObjectResult("Lead created successfully"); }

**Sample Azure Function (Python):**

import azure.functions as func import json

def main(req: func.HttpRequest) -> func.HttpResponse: req\_body = req.get\_json() # Validate data, push to CE, handle errors return func.HttpResponse("Lead created successfully", mimetype="text/plain")

## 10. Business Rules

* **Mandatory Fields Rule:** Make First Name, Last Name, Email, Company Name, Lead Source required if Status=New.
* **Block Conversion:** Prevent conversion if mandatory fields are missing or email invalid.
* **Disallow Duplicate:** Show error and block save if duplicate detected.

**Business Rule Logic Example:**

* If Status=New and Company Name is empty â†’ Show error "Company Name is required."
* If conversion initiated but Email is missing â†’ Block conversion and prompt user.

## 11. Business Process Flow

**Lead Qualification Process Stages:**

1. Lead Entry
2. Data Validation
3. Lead Assignment
4. Qualification/Disqualification
5. Conversion (to Account/Contact/Opportunity)

## 12. Actions

* **Custom Actions:** Not needed for lead conversion, as OOB process is sufficient.

## 13. Ribbon Workbench Customizations

* **Enable Rule:** Show â€œConvertâ€ button only if lead passes validation and is not a duplicate.  
  **Display Rule:** Hide â€œConvertâ€ button for already qualified/disqualified leads.  
  **JavaScript Reference:** Use *validateLeadRequiredFields()* for enable rule.

## 14. Power BI Reports

* Lead conversion rate by source and territory
* Average time to conversion
* Duplicate leads trends

## 15. Power Automate Workflows

* **Type:** Automated cloud flow
* **Design Steps:**
  1. Trigger: When a new lead is created (from web, API, or manual)
  2. Condition: If lead source is "Web" and status is "New", send alert to inside sales queue
  3. Action: Assign lead to sales rep as per assignment logic
  4. Action: Send email notification and post to Teams

## 16. Integration Recommendations

* Web form/API integration for lead creation via Azure Functions and Logic Apps
* Bulk import/export using Data Import Wizard or Power Automate
* Duplicate check using custom plugin or Microsoftâ€™s built-in duplicate detection
* Power BI for reporting and analytics

Â©2025 AI-Generated Dynamics 365 CE Technical Specification  
Generated for reference only. Validate before implementation.