📊 Job Application Tracker - Data Dictionary

> **"Complete Schema Reference for Team Collaboration"**

🎯 Team Alignment Document

This data dictionary serves as our single source of truth for the custom objects and fields in our Job Application Tracker. Please review, critique, and suggest improvements so we can build together effectively!

🏛 Custom Object Overview

Job_Application__c (Primary Custom Object)

Purpose: Track the complete lifecycle of job applications from discovery to final outcome
Sharing Model: Private (users see only their own applications)
Record Name: Auto-number format "JA-{0000}"

📋 Complete Field Specifications

� Field Summary

- **Total Custom Fields**: 23 fields
- **Required Fields**: 2 (Company_Name__c, Position_Title__c, Status__c)
- **Currency Fields**: 9 (salary and tax calculations)
- **Picklist Fields**: 3 (Status__c, Location__c, Rating__c)
- **Lookup Fields**: 1 (Primary_Contact__c)
- **Validation Rules**: 3 active rules
- **Auto-Number**: Job Application Name (JA-{0000})

� 🔍 Core Identification Fields

Field Name	API Name	Type	Length	Required	Description
Job Application Name	Name	Auto Number	-	☑	Auto-generated: "JA-{0000}"
Company Name	Company_Name__c	Text	255	☑	Name of the company/organization
Position Title	Position_Title__c	Text	255	☑	Job title or role name (Label: "Position/Title")
Description	Description__c	Long Text Area	32,768	✘	Detailed job description and requirements

🗓 Timeline & Status Fields

Field Name	API Name	Type	Options/Format	Required	Description

Field Name	API Name	Type	Options/Format	Required	Description
Application Date	Application_Date__c	Date	-	✗	Date when application was submitted
Status	Status__c	Picklist	See status values below	☑	Current stage of application
Follow-up Date	Follow_Up_Date__c	Date	-	✗	Next planned follow-up date

Status Picklist Values (Default: Saved):

- Saved - Job found but not yet applied (DEFAULT)
- Applying - In process of applying
- Applied - Application submitted
- Interviewing - In interview process
- Negotiating - Discussing offer terms
- Accepted - Offer accepted
- Closed - Application closed/rejected

💰 Compensation & Tax Fields

Field Name	API Name	Type	Format	Required	Description
Salary	Salary__c	Currency	$0.00	✗	Annual salary amount
Federal Tax	Federal_Tax__c	Currency	$0.00	✗	Calculated federal tax (yearly)
Social Security Tax	Social_Security_Tax__c	Currency	$0.00	✗	Calculated Social Security tax (yearly)
Medicare Tax	Medicare_Tax__c	Currency	$0.00	✗	Calculated Medicare tax (yearly)
Take Home Pay (Yearly)	Take_Home_Pay_Yearly__c	Currency	$0.00	✗	Calculated after-tax amount (yearly)
Take Home Pay (Monthly)	Take_Home_Pay_Monthly__c	Currency	$0.00	✗	Calculated after-tax amount (monthly)

📍 Location & Reference Fields

Field Name	API Name	Type	Options	Required	Description
Location	Location__c	Picklist	Remote/Hybrid/On-site	✖	Work arrangement type
Job URL	Job_URL__c	URL	-	✖	Link to original job posting
Primary Contact	Primary_Contact__c	Lookup	Contact object	✖	Reference to Contact record

� Analytics & Market Data Fields

Field Name	API Name	Type	Format	Required	Description
Market Salary Min	Market_Salary_Min__c	Currency	$0.00	✖	Market research - minimum salary
Market Salary Max	Market_Salary_Max__c	Currency	$0.00	✖	Market research - maximum salary
Market Salary Median	Market_Salary_Median__c	Currency	$0.00	✖	Market research - median salary
Salary Competitiveness	Salary_Competitiveness__c	Text	-	✖	Calculated competitiveness rating
Market Analysis Date	Market_Analysis_Date__c	Date	-	✖	When market analysis was performed

📝 Notes & Rating Fields

Field Name	API Name	Type	Options	Required	Description
Notes	Notes__c	Long Text Area	32,768 chars	✖	General notes and observations
Rating	Rating__c	Picklist	1-5 scale	✖	Interest level rating

Rating Picklist Values:

- 1 - 1 - Low Interest
- 2 - 2 - Some Interest
- 3 - 3 - Moderate Interest

- **4** - 4 - High Interest
- **5** - 5 - Very High Interest

🔒 Security & Permissions Model

Object-Level Security

- **Private Sharing Model**: Users can only see their own job applications
- **Organization-Wide Defaults**: Private
- **Role Hierarchy**: Respects hierarchy (managers can see subordinates' data)

Field-Level Security Recommendations

Field Category	Manager Access	User Access	Viewer Access
Core Fields (Name, Company, Position)	Read/Edit	Read/Edit	Read Only
Salary Fields	Read/Edit	Read/Edit	Hidden
Contact Information	Read/Edit	Read/Edit	Read Only
Notes & Documentation	Read/Edit	Read/Edit	Read Only
System Fields (Created, Modified)	Read Only	Read Only	Read Only

Permission Sets

1. **Job_Application_Manager**: Full access to all fields and records
2. **Job_Application_User**: Standard user access (own records only)
3. **Job_Application_Viewer**: Read-only access for reporting

---

🔄 Automation & Business Logic

Validation Rules

1. Salary Range Validation

```
Rule Name: Salary_Range_Validation
Error Condition: AND(
    NOT(ISBLANK(Salary__c)),
    OR(
        Salary__c < 20000,
        Salary__c > 1000000,
        AND(
            ISPICKVAL(Status__c, "Applied"),
            Salary__c > 500000
        )
    )
)
Error Message: "Salary must be between $20,000 and $1,000,000. For Applied status,
```

```
salary cannot exceed $500,000 without approval."
Error Location: Salary__c
```

2. Application Date Validation

```
Rule Name: Application_Date_Validation
Error Condition: AND(
    NOT(ISBLANK(Application_Date__c)),
    OR(
        Application_Date__c > TODAY(),
        Application_Date__c < TODAY() - 365
    )
)
Error Message: "Application date cannot be in the future or more than 1 year in
the past."
Error Location: Application_Date__c
```

3. Status Progression Validation

```
Rule Name: Status_Progression_Validation
Error Condition: AND(
    ISCHANGED(Status__c),
    OR(
        AND(
            ISPICKVAL(PRIORVALUE(Status__c), "Closed"),
            NOT(ISPICKVAL(Status__c, "Applied"))
        ),
        AND(
            ISPICKVAL(PRIORVALUE(Status__c), "Accepted"),
            NOT(ISPICKVAL(Status__c, "Closed"))
        ),
        AND(
            ISPICKVAL(Status__c, "Negotiating"),
            NOT(ISPICKVAL(PRIORVALUE(Status__c), "Interviewing"))
        )
    )
)
Error Message: "Invalid status change. Please follow proper progression: Applied →
Interviewing → Negotiating → Accepted/Closed. Closed applications can only be
reopened to Applied status."
Error Location: Status__c
```

Trigger Automation

- **Status Change Tasks**: Automatically create tasks when status changes
- **Take Home Pay Calculation**: Calculate after-tax pay when salary is entered
- **Follow-up Reminders**: Set follow-up dates based on status

📊 Reporting & Analytics Fields

Formula Fields (Calculated)

Field Name	API Name	Return Type	Formula Purpose
Days Since Application	`Days_Since_Application__c`	Number	`TODAY() - Application_Date__c`
Salary Range Midpoint	`Salary_Range_Midpoint__c`	Currency	`(Salary_Range_Min__c + Salary_Range_Max__c) / 2`
Application Age Category	`Application_Age_Category__c`	Text	Categorizes applications by age

Rollup Summary Fields (Future Enhancement)

Note: These would require a junction object or custom development

- Total Applications by Company
- Average Salary by Industry
- Success Rate by Source

🎯 Team Discussion Points

🤯 Questions for Team Review

1. **Field Completeness**: Are we missing any critical fields for tracking job applications?

2. **Picklist Values**: Do the Status picklist values cover all scenarios we might encounter?

3. **Field Types**: Are the field types and lengths appropriate for our use cases?

4. **Security Model**: Does the proposed security model meet our requirements?

5. **Automation Scope**: What additional automation would be valuable?

🔄 Potential Enhancements

1. **Industry Field**: Add picklist for job industry/sector
2. **Remote Work Options**: Add field for remote/hybrid/onsite
3. **Application Method**: Track how application was submitted
4. **Rejection Reason**: Capture feedback when rejected
5. **Skills Match**: Track specific skills alignment

🏛️ Future Object Considerations

1. **Interview__c**: Separate object for multiple interview rounds
2. **Contact__c**: Separate object for networking contacts

3. **Company__c**: Master data object for company information
4. **Skill__c**: Skills tracking and matching

📋 Implementation Checklist

Phase 1: Core Object ☑

- ☑ Create Job_Application__c custom object
- ☑ Add all core fields
- ☑ Configure field-level security
- ☑ Set up validation rules

Phase 2: Automation 🔄

- ☐ Implement trigger for status-based tasks
- ☐ Add salary calculation logic
- ☐ Create workflow rules for follow-ups

Phase 3: User Experience 📋

- ☐ Design page layouts
- ☐ Create list views
- ☐ Build Lightning Web Components

Phase 4: Reporting 📊

- ☐ Create standard reports
- ☐ Build analytics dashboard
- ☐ Implement KPI tracking

🤝 Team Collaboration Notes

Git Branch Strategy

- **main**: Production-ready code
- **develop**: Integration branch
- **feature/[name]**: Individual feature branches
- **[teammate-name]/[feature]**: Personal development branches

Naming Conventions

- **API Names**: Use descriptive names with proper suffixes (__c)
- **Field Labels**: User-friendly, consistent terminology
- **Validation Rules**: Descriptive names indicating purpose
- **Automation**: Clear naming for triggers and workflows

Code Review Focus Areas

1. **Data Model Consistency**: Ensure fields align with business requirements

2. **Security Implementation**: Verify proper access controls

3. **Performance Considerations**: Optimize for scale

4. **User Experience**: Ensure intuitive field placement and labels

---

💬 **Feedback Request**

Team, please review and provide feedback on:

1. **Missing Fields**: What additional fields should we consider?

2. **Field Types**: Are the chosen field types optimal?

3. **Picklist Values**: Do we need additional status values or other picklist options?

4. **Security Model**: Does the proposed security approach meet our needs?

5. **Automation Ideas**: What business logic should we automate?

6. **Naming Conventions**: Are the API names clear and consistent?

Please share your thoughts so we can finalize our data model before moving to implementation! 🚀

Next Steps: Once we align on the data dictionary, we'll move to implementing the automation layer and user interface components. This foundation will ensure we're all building on the same solid base! 🤝