Development: Data Management

Lesson 19: Data Management Overview

Application Building Blocks

**Applications
Tabs
Page Layouts
Record Types**

User Interface

- Force.com Pages
- Web controls
- Sites

- Workflow
- Validation Rules
- Approval Processes

Business Logic

- Force.com Page Controllers
- Force.com code
- Web Services API

- Objects
- Fields
- Relationships

Data Model

- Web Services API
- Metadata API

Declarative

Programmatic

Simplicity + Speed

Control + Flexibility

2

Lesson Objectives

By the end of this lesson, you will be able to :

- List typical data management operations
- List the ways to obtain record IDs
- State typical use case for inserting system fields
- Locate resources explaining algorithm for converting 15-digit IDs to 18-digit form
- Explain typical situation where the use of the 15-digit ID can cause problems

Exporting Data

- Get reference IDs for inserts/updates
- Download records for backup

Inserting Data

- Load Users
- Load existing Positions

Updating Data

- De-duplicate existing data
- Introduce Record Types

Upserting Data

- Migrate new and existing records from a legacy system
- Keep Salesforce in sync with another systems

Deleting Data

- Free up space used by too much legacy data
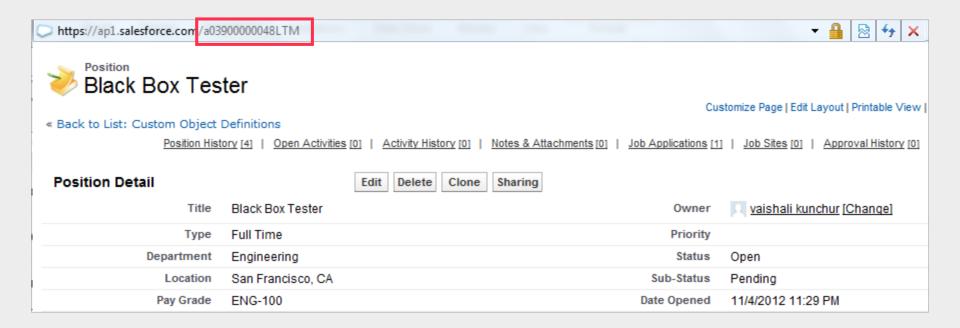- Fix mistakes!

Unique identifier of a record

Analogous to a primary key or foreign key field in a database table

Salesforce generates an ID value when a new record is created (e.g., a00D0000005iTiZ)

19.2: Record IDs
Where to get Salesforce IDs?

IDs may be obtained in the following ways:

1. URL
2. Report
3. Web Service API (e.g., the Data Loader)
4. Formulas

19.2: Record IDs
Format for Record IDs

Salesforce.com Object IDs come in two forms:
- 15-digit case-sensitive form
- 18-digit case-insensitive form

Reports and Object IDs
- Reports (and Office Edition) return 15-digit IDs
- Report framework does not expose IDs for all objects

API and Object IDs
- API always returns 18-digit IDs
- API will accept either the 15-digit or 18-digit format

19.3: Object Relationships
Object Relationships

Relationships exist between objects. For example:
- All Positions have an owner
- Candidates are related to Position through a Job Application
- Reviews are associated to a Job Application

Relationships are expressed through:
- Related lists and lookup in the application
- IDs (foreign keys) in the database

19.3: Object Relationships
Determine the Order to Load Data

Object relationships introduce data dependencies
Dependencies dictate the order of data load
To load Review data, you must:

1. Load Users
2. Load Positions
3. Load Candidates
4. Load Job Applications
5. Load Reviews

Inserting System Fields

Allows you to set Created Data, Created By, Last Update Date, Last Update By fields on initial creation of records

Only accessible through API

Backwards compatible with all SOAP-based APIs

Works for all Custom Objects

Restricted to Account, Opportunity, Contact, Lead, Case, Task, and Event Standard Objects

Must be enabled by salesforce.com Customer Support

Best Practice: enable only for initial data migration

Summary

Essential Data Management Operations – Insert, Update, Delete, Upsert

Record IDs

Object Relationships - Determine the Order to Load Data

Inserting System Fields

Name the ways to obtain a record's ID.

Why can it be dangerous to work with the 15-digit case-sensitive form of record ID?

The Created Date can be updated for a record via the API as long as the Insertable System Fields feature is turned on.

> True or False