REVBLACK

SALESFORCE AUDIT CHECKLIST

What to check in your Salesforce org, in what order, with every query ready to run.

WHO IT'S FOR

Salesforce admins and RevOps leads, plus the CRO or CMO who inherited the org

WHAT YOU NEED

System Administrator access, the Developer Console Query Editor, and a short talk with the people who use Salesforce every day

FULL GUIDE

revblack.com/guides/salesforce-audit-checklist

How to use this checklist

Work through the 15 steps in order. Each one starts with what you'll have when it's done, so you know what you're working toward. Tick each box as you go and write the counts straight onto the page.

To run a query, open the Developer Console (gear icon, then Developer Console), go to the Query Editor tab, paste the query and click Execute. Queries marked TOOLING API need the Use Tooling API box ticked first. Copy queries from the full guide if you'd rather not retype them.

Treat the whole audit as read only. Record what you find, and change nothing until you've walked the team through the findings.

1 **Before you open Salesforce**
Steps 1 to 3. Set the objective, map how the team should work, get the right access and tools.

2 **Salesforce system checks**
Steps 4 to 12. Licenses, security, packages, objects, data quality, duplicates, automation and reporting.

3 **Reviewing the findings**
Steps 13 to 15. Test each pain point, work out the cause with the team, write it up.

Some checks depend on your Salesforce edition and licenses. The full guide notes which ones, with links to related RevBlack guides.

PART 1 · STEPS 1 TO 3

BEFORE YOU OPEN SALESFORCE

Talk to the team before you open Setup. Know what sales and marketing actually do in Salesforce before you judge what's in it.

1 Decide what the audit is for

> **When this step is done, you'll have** a short written goal for the audit, plus a list of pain points with the name of the person who raised each one.

- [ ] Write down the question the audit needs to answer, such as whether the pipeline can be trusted for forecasting.
- [ ] Talk to the people who use Salesforce every day: reps, sales managers, and whoever runs marketing or service in it.
- [ ] Write the notes from those conversations yourself rather than relying on AI notes, so you understand each pain point before you test it.
- [ ] Log each pain point with who raised it. You'll test every one in step 13.
- [ ] Turn each complaint into something you can check in the org.

THE AUDIT QUESTION

PAIN POINTS

Pain point	Raised by	What you'll check

2 Map the processes Salesforce should mirror

When this step is done, you'll have a written map of how the team is supposed to work: the opportunity stages with their entry and exit criteria, the rules Salesforce actually enforces, and what your team means by a lead and a contact.

- [ ] List the opportunity stages and the entry and exit criteria the team agreed on.
- [ ] Note which rules Salesforce enforces (validation rules, required fields, record types) and which exist only on paper.
- [ ] Write down what counts as a lead and what counts as a contact. We recommend retiring the Lead object and tracking early interest as a pre-pipeline opportunity.

3 Confirm your access and tools

When this step is done, you'll have System Administrator access for as long as the audit runs, and a way to run SOQL queries that you've tested on one simple query.

- [ ] You have System Administrator access, or a permission set with View Setup and Configuration and View All Data.
- [ ] You can run a query: gear icon > Developer Console > Query Editor tab, paste the query, click Execute.
  Data Loader and the Salesforce Inspector Reloaded Chrome extension also run SOQL.
- [ ] You know where the Use Tooling API box is. Queries marked Tooling API in this checklist need it ticked.
- [ ] The audit is read only. No changes until the team has seen the findings and why each change is needed.
- [ ] If HubSpot is connected, you've booked time with whoever owns the sync.

PART 2 · STEPS 4 TO 12

SALESFORCE SYSTEM CHECKS

Most checks are a SOQL query that returns a count. Copy each query below into the Developer Console Query Editor, click Execute, and write the number on the line next to it. Queries marked Tooling API need the Use Tooling API box ticked.

4 Licenses, storage and limits

When this step is done, you'll have the org's edition, licenses in use against licenses paid for, the licenses still assigned to people who've left, and data and file storage against the limit.

- ☐ Edition _____
- ☐ User licenses in use / paid _____
- ☐ Licenses held by people who left _____
- ☐ Data storage used / limit _____
- ☐ File storage used / limit _____
- ☐ API usage vs limit _____
- ☐ Setup > Quick Find: Company Information, Storage Usage and System Overview.
- ☐ Note which record types and which users take up the most storage.

5 Users, permissions and security

When this step is done, you'll have the System Administrators listed by name, each profile with its active user count, the Health Check score with the high-risk settings it flags, whether multi-factor authentication is in place, and who has changed the org's settings recently.

☐ **System Administrators (list)** _____

```
SELECT Name, Username, IsActive FROM User WHERE Profile.Name = 'System Administrator' ORDER BY IsActive DESC, Name
```

☐ **Active users per profile** _____

```
SELECT Profile.Name, COUNT(Id) FROM User WHERE IsActive = true GROUP BY Profile.Name ORDER BY COUNT(Id) DESC
```

☐ **Users per permission set** _____

```
SELECT PermissionSet.Name, COUNT(AssigneeId) FROM PermissionSetAssignment WHERE PermissionSet.IsOwnedByProfile = false GROUP BY PermissionSet.Name ORDER BY COUNT(AssigneeId) DESC
```

☐ **Latest 50 setup changes** _____

```
SELECT CreatedDate, CreatedBy.Name, Section, Display FROM SetupAuditTrail ORDER BY CreatedDate DESC LIMIT 50
```

☐ **Lead and case assignment rules** _____

```
SELECT Name, SobjectType, Active FROM AssignmentRule
```

- [ ] Health Check score _____ [ ] High-risk settings flagged _____
- [ ] Multi-factor authentication is in place for all logins to the user interface.
- [ ] Org-wide defaults in Sharing Settings: note any object more open than the business expects.
- [ ] Profiles restricting login hours or login IP ranges: flag gaps where the business expects them.
- [ ] View Setup Audit Trail: who has been changing the configuration (at least the last 180 days are kept).

6 Packages, integrations and sandboxes

When this step is done, you'll have every installed package grouped by what it does, every integration that writes to Salesforce with the user it connects as, and the sandboxes the team uses to test changes.

- [ ] Installed packages TOOLING API _____

```
SELECT SubscriberPackage.Name, SubscriberPackage.NamespacePrefix, SubscriberPackageVersion.Name FROM InstalledSubscriberPackage
```

INTEGRATIONS THAT WRITE TO SALESFORCE

Integration	User it connects as	Records it creates or updates

- [ ] Sandboxes (and types) _____ [ ] Sandbox licenses used / available _____
- [ ] Flag any package nobody can explain or still uses.
- [ ] Changes are built and tested in a sandbox before they reach production.
- [ ] If HubSpot is connected, review the sync from the HubSpot side too (HubSpot audit checklist, step 9).

7 Objects and fields

When this step is done, you'll have the custom objects your team built with what each one is for and whether it's documented, plus field, record type and validation rule counts for each key object.

Run these for Opportunity, then swap in Account, Contact, Lead and Case. The two field queries list the fields, so read the total rows above the results.

- [ ] All fields _____

```
SELECT QualifiedApiName FROM FieldDefinition WHERE EntityDefinition.QualifiedApiName = 'Opportunity'
```

- [ ] Custom fields

```
SELECT QualifiedApiName FROM FieldDefinition WHERE EntityDefinition.QualifiedApiName = 'Opportunity' AND QualifiedApiName LIKE '%\_\_c'
```

- [ ] Record types

```
SELECT COUNT() FROM RecordType WHERE SobjectType = 'Opportunity'
```

- [ ] Validation rules TOOLING API

```
SELECT COUNT() FROM ValidationRule WHERE EntityDefinition.QualifiedApiName = 'Opportunity'
```

CUSTOM OBJECTS YOUR TEAM BUILT (NO NAMESPACE PREFIX)

Custom object	What it's for	Documented?	Could a standard object do it?

- [ ] Open the record pages for account, contact, opportunity, lead and case the way a rep does. Note cluttered layouts.
- [ ] Use Where is this used? before anyone proposes deleting a field.
- [ ] Compare the opportunity stages with the criteria from step 2.

8 Data quality by query

> **When this step is done, you'll have** a count written next to every query below. Together they're the data quality readout.

CONTACTS

- [ ] Total contacts

```
SELECT COUNT(Id) FROM Contact
```

- [ ] No email

```
SELECT COUNT(Id) FROM Contact WHERE Email = null
```

- [ ] No first name

```
SELECT COUNT(Id) FROM Contact WHERE FirstName = null
```

- [ ] Inactive owner

```
SELECT COUNT(Id) FROM Contact WHERE Owner.IsActive = false
```

- [ ] Bounced email

```
SELECT COUNT(Id) FROM Contact WHERE EmailBouncedDate != null
```

ACCOUNTS

- [ ] Total accounts

```
SELECT COUNT(Id) FROM Account
```

- [ ] No contacts

```
SELECT COUNT(Id) FROM Account WHERE Id NOT IN (SELECT AccountId FROM Contact WHERE AccountId != null)
```

- [ ] No opportunities

```
SELECT COUNT(Id) FROM Account WHERE Id NOT IN (SELECT AccountId FROM Opportunity WHERE AccountId != null)
```

- [ ] No industry

```
SELECT COUNT(Id) FROM Account WHERE Industry = null
```

- [ ] No website

```
SELECT COUNT(Id) FROM Account WHERE Website = null
```

- [ ] Inactive owner

```
SELECT COUNT(Id) FROM Account WHERE Owner.IsActive = false
```

OPPORTUNITIES

- [ ] Total opportunities

```
SELECT COUNT(Id) FROM Opportunity
```

- [ ] No amount

```
SELECT COUNT(Id) FROM Opportunity WHERE Amount = null
```

- [ ] No type

```
SELECT COUNT(Id) FROM Opportunity WHERE Type = null
```

- [ ] No lead source

```
SELECT COUNT(Id) FROM Opportunity WHERE LeadSource = null
```

- [ ] No contact roles

```
SELECT COUNT(Id) FROM Opportunity WHERE Id NOT IN (SELECT OpportunityId FROM OpportunityContactRole)
```

- [ ] Open, no next step

```
SELECT COUNT(Id) FROM Opportunity WHERE IsClosed = false AND NextStep = null
```

- [ ] Open, close date passed

```
SELECT COUNT(Id) FROM Opportunity WHERE IsClosed = false AND CloseDate < TODAY
```

- [ ] Open, untouched 60+ days

```
SELECT COUNT(Id) FROM Opportunity WHERE IsClosed = false AND LastModifiedDate < LAST_N_DAYS:60
```

- [ ] Open, inactive owner

```
SELECT COUNT(Id) FROM Opportunity WHERE IsClosed = false AND Owner.IsActive = false
```

LEADS

- [ ] Total leads

```
SELECT COUNT(Id) FROM Lead
```

- [ ] No email

```
SELECT COUNT(Id) FROM Lead WHERE Email = null
```

- [ ] No first name

```
SELECT COUNT(Id) FROM Lead WHERE FirstName = null
```

- [ ] No lead source

```
SELECT COUNT(Id) FROM Lead WHERE LeadSource = null
```

- [ ] Inactive owner

```
SELECT COUNT(Id) FROM Lead WHERE Owner.IsActive = false
```

CAMPAIGNS AND CASES

- [ ] Campaigns missing a date

```
SELECT COUNT(Id) FROM Campaign WHERE StartDate = null OR EndDate = null
```

- [ ] Campaigns with no members

```
SELECT COUNT(Id) FROM Campaign WHERE Id NOT IN (SELECT CampaignId FROM CampaignMember)
```

- [ ] Active after end date

```
SELECT COUNT(Id) FROM Campaign WHERE IsActive = true AND EndDate < TODAY
```

- [ ] Campaigns, inactive owner

```
SELECT COUNT(Id) FROM Campaign WHERE Owner.IsActive = false
```

- [ ] Cases with no subject

```
SELECT COUNT(Id) FROM Case WHERE Subject = null
```

- [ ] Cases, inactive owner

```
SELECT COUNT(Id) FROM Case WHERE Owner.IsActive = false
```

- [ ] Open cases 60+ days old

```
SELECT COUNT(Id) FROM Case WHERE IsClosed = false AND CreatedDate < LAST_N_DAYS:60
```

The opportunity counts matter most: they tell you whether the pipeline can be trusted for forecasting.

9 Duplicates

When this step is done, you'll have the active duplicate and matching rules with what each one blocks or allows, plus duplicate counts on the fields that matter to you and a few example records saved as evidence.

- [ ] Note which duplicate and matching rules are active, what they block or allow, and whether anyone can explain why.

Each query lists values that appear on more than one record. Write down how many rows come back.

- [ ] **Duplicate contact emails** _____

```
SELECT Email, COUNT(Id) FROM Contact WHERE Email != null GROUP BY Email HAVING COUNT(Id) > 1
```

- [ ] **Duplicate lead emails** _____

```
SELECT Email, COUNT(Id) FROM Lead WHERE Email != null GROUP BY Email HAVING COUNT(Id) > 1
```

- [ ] **Duplicate account names** _____

```
SELECT Name, COUNT(Id) FROM Account GROUP BY Name HAVING COUNT(Id) > 1
```

- [ ] **Duplicate account websites** _____

```
SELECT Website, COUNT(Id) FROM Account WHERE Website != null GROUP BY Website HAVING COUNT(Id) > 1
```

- [ ] Save a few example records as evidence. Hold off on merging until the findings are reviewed.

A duplicate query that comes back with 2,000 rows has hit the limit for one query, and that alone tells you the problem is serious.

10 Leads and contacts against your definitions

When this step is done, you'll have a short list of the places where the team's actual use of leads and contacts differs from the definitions you agreed in step 2.

- [ ] Look for the same person sitting in both objects, and for leads that never get converted.
- [ ] If HubSpot is connected, check whether new HubSpot contacts arrive in Salesforce as leads or as contacts.

Where use differs from the definition	Example

11 Flows, legacy automation and Apex

When this step is done, you'll have counts of active flows by type, active Workflow Rules and Process Builder processes, and custom Apex classes and triggers, plus a list of automations that look duplicated, conflict with each other, or that nobody can explain.

- [ ] **Active flows by type** TOOLING API _____

```
SELECT ProcessType, COUNT(Id) FROM Flow WHERE Status = 'Active' GROUP BY ProcessType
```

- [ ] **Record-triggered flows by object** _____

```
SELECT Label, TriggerObjectOrEventLabel, TriggerType FROM FlowDefinitionView WHERE IsActive = true AND
TriggerObjectOrEventLabel != null ORDER BY TriggerObjectOrEventLabel
```

- [ ] **Custom Apex classes** _____

```
SELECT COUNT() FROM ApexClass WHERE NamespacePrefix = null
```

- [ ] **Custom Apex triggers** _____

```
SELECT COUNT() FROM ApexTrigger WHERE NamespacePrefix = null
```

- [ ] **Org-wide code coverage (%)** TOOLING API _____

```
SELECT PercentCovered FROM ApexOrgWideCoverage
```

- [ ] Active Workflow Rules _____ [ ] Active Process Builder processes _____
- [ ] Flag flows that look duplicated, conflict with each other, or that nobody can explain.
- [ ] Paused and Failed Flow Interviews: note any failure that keeps repeating.
- [ ] Workflow Rules and Process Builder lost support on December 31, 2025. Treat them as a migration item, not an emergency.
- [ ] Coverage below 75% means the next Apex deployment could fail.

12 History tracking, activity capture and reporting

> **When this step is done, you'll have** the fields with history tracking turned on, whether the org has Field Audit Trail, how captured emails and events are stored, and whether the main leadership reports still run on fields the team fills in.

- [ ] Field history tracking is on for stage, amount and owner on opportunities (up to 20 fields per object).
- [ ] Field Audit Trail: listed under Permission Set Licenses or Feature Licenses in Company Information?
  Yes / No
- [ ] Einstein Activity Capture: internal domain settings, contact sync settings, inactive configurations, and whether emails are saved as activity records.
- [ ] Sales Engagement Settings: turned on, and still used?
- [ ] Reporting Snapshots: list the active ones and confirm someone still reads them.
- [ ] Einstein Lead Scoring: on, and still matching how sales qualifies leads? If it's off, check the license before suggesting it.
- [ ] The main leadership reports and dashboards still run on fields the team fills in.

PART 3 · STEPS 13 TO 15

REVIEWING THE FINDINGS

The counts and settings show what's in the org. This part is where you find out why it got that way, with the people who use it.

13 Check every pain point against the evidence

When this step is done, you'll have a short note for each pain point from step 1 saying what you found, with evidence such as query results, screenshots or example records.

Pain point (from step 1)	What you found	Evidence

14 Work out the cause with the team

When this step is done, you'll have one question for the team about each critical finding, their answers, and the findings ranked by business impact.

A finding can be a technical problem, a process problem or an adoption problem. Check before you assume.

Critical finding	Question for the team	Their answer	Rank

15 Write up the audit

When this step is done, you'll have the finished audit write-up, ready to walk the team through.

- [ ] A readout of the org: the counts and settings from steps 4 to 12.
- [ ] Critical findings, ranked by business impact, such as a high duplicate count or undocumented custom objects.
- [ ] Questions for the team, from step 14.

Check that it worked

Before you call the audit finished, check that:

- [ ] You understand every finding well enough to explain it to the team in your own words.
- [ ] Every key object, custom object and setting covered in the steps has been reviewed.
- [ ] Every pain point from step 1 has a finding or an explanation.
- [ ] Every critical finding has a question for the team.
- [ ] No section is left blank or marked 'to be done' without a reason.

Re-run the core queries quarterly, or whenever you take over the org or the team stops trusting the data.

REVBLACK

RATHER HAVE SOMEONE ELSE RUN IT?

RevBlack runs Salesforce and HubSpot audits for B2B teams.

Book a Salesforce audit call at revblack.com/contact-us