SAIMA

Salesforce Architects In The Making

Agenda

1. Welcome & Introduction

2. Implicit Sharing
3. Skews and Record Locking

4. Summary

Who We Are

Johann Furmann
Technical Product
Manager Salesforce

jfuermann@gmail.com

Nora Nicklis
Freelance Salesforce
Trainer & Consultant

nora@nicklis.de

What We Want To Achieve

We want to **support each other** in **preparing for the Technical Architect certification** by:

- Creating and sharing content that helps to prepare for the certification
- Facilitating small study groups to make room for discussion based learning

If you have more ideas on what we could/should do, please let us know!

Housekeeping Rules

- Post your questions in the chat, we will answer them during or after the session

- We are not perfect, neither do we know everything. If you spot a mistake, please do let us know!

- Interested in presenting? Please reach out to us, everybody is welcome to present!

Session 1

Implicit Sharing

Johann Furmann

SAIMA Group

17.04.2020

@fliack

Kind of Sure

Agenda

- Parent
- Child
- Portal
- Case
- Share Groups (High Volume)

Parent
Definition

Read-only access to the parent account for a user with access to a child record

- Not used when sharing on the child is controlled by its parent
- Expensive to maintain with many account children
- When a user loses access to a child, Salesforce needs to check all other children to see if it can delete the implicit parent.

Parent
Explanation

- Only affects Account to Case, Contact and Opportunity
- Account = private AND Child != “Controlled by Parent”

As soon as a User has at least Read access to a Case, Contact or Opportunity the User has Read access to the related Account.

Kind of Sure

Parent
Demo

Demo User: Carla Sunflower (Salesforce License)

OWD Private: Account, Contact, Case, Opportunity

1. Carla Sunflower no access to any Accounts, Cases, Contacts or Opportunities
2. Man. Share a Contact with Carla Sunflower

-> Carla Sunflower gets Read access to Account

Child
Definition

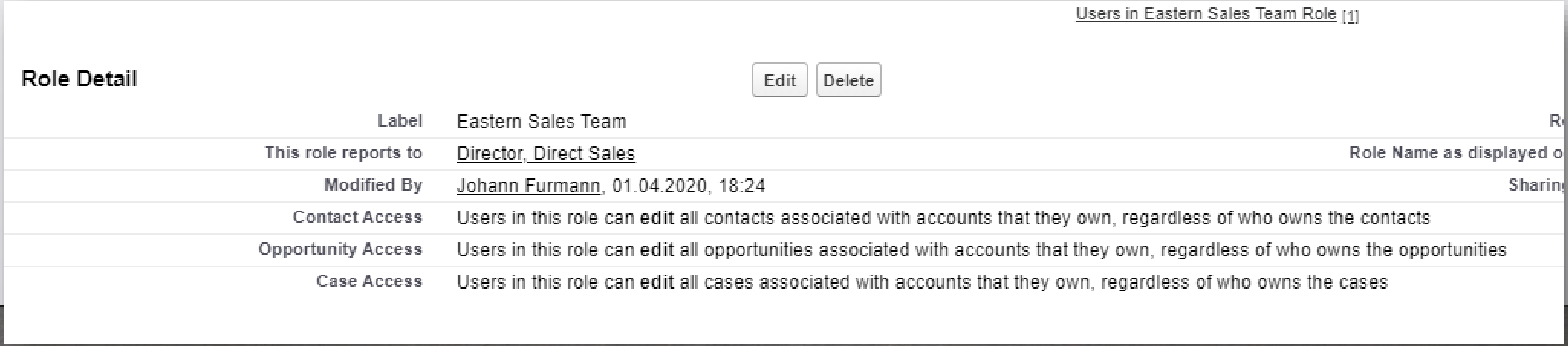
Access to child records for the owner of the parent account

- Not used when sharing on the child is controlled by its parent
- Controlled by child access settings for the account owner's role
- Supports account sharing rules that grant child record access
- Supports account team access based on team settings
- When a user loses access to the parent, Salesforce needs to remove all the implicit children for that user.

Child
Explanation

- Only affects Account to Case, Contact and Opportunity
- Only affects Account Owner

Each role independently can define what Child-Record access the Account Ownership provides.



Role Detail

Label	Eastern Sales Team	R
This role reports to	Director, Direct Sales	Role Name as displayed o
Modified By	Johann Furmann, 01.04.2020, 18:24	Sharin
Contact Access	Users in this role can **edit** all contacts associated with accounts that they own, regardless of who owns the contacts	
Opportunity Access	Users in this role can **edit** all opportunities associated with accounts that they own, regardless of who owns the opportunities	
Case Access	Users in this role can **edit** all cases associated with accounts that they own, regardless of who owns the cases	

Child
Demo

Demo User: Carla Sunflower (Role: Eastern Sales Team)

OWD Private: Account, Contact, Case, Opportunity

1. Set Carla Sunflower as Account Owner

-> Carla Sunflower gets Edit Access to all Opportunities

2. Setting “Opportunity Access” is “Cannot access” for Role

-> Carla can not Access the Opportunity anymore

Portal
Definition

Access to portal account and all associated contacts for all portal users under that account.

Shared to the lowest role under the portal account

Portal
Explanation

- Only Partner Community License
- Only affects Contacts which were enabled for Partner Community

A Partner-Community User has read access to her Account and all other Contacts on that Account.

Portal
Demo

Demo Contact: Mia Learnsalot

OWD Private: Account & Contact

1. Log-In to community

-> Her Account and other Contacts visible

Case Access
Definition

Case access—If a portal or customer community plus user is a contact on a case, then the user has Read and Write access on the case.

Case Access
Explanation

The Contact for a Case has Read & Edit Access on that Case.

- Only affects Partner and Community Plus Licenses
- Community License Users need Sharing-Set

Kind of Sure

Case Access *Demo*

Demo Contact / Customer: Mia Learnsalot

OWD Private: Account, Contact and Case

1. Mia sees no Cases
2. Create a Case for Mia
3. Add Mia as the Contact to the Case

-> Mia can see and edit the Case

Share Group (High Volume)
Definition

Access to data owned by high volume users associated with a sharing set for users member of the sharing set's access group.

All members of the sharing set access group gain access to every record owned by every high volume user associated with that sharing set.

Share Group (High Volume)
Explanation

Profile: Customer Community

- Record Ownership by User is necessary
- Affects (almost?) all Objects
- Record Owner needs profile in Share Set
- Any Record owned by a Profile mentioned in the Share Set is shared with any user mentioned in the related Share Group

Kind of Sure

Share Group (High Volume) *Demo*

High Volume Community User AND Carla Sunflower

1. Create share set for *Custom Demo Object* and Community Profile
2. Share Group add Carla Sunflower
3. High Volume Community User creates Case Record

-> Record is shared to Carla Sunflower too

Implicit Sharing

Johann Furmann

SAIMA Group

17.04.2020

@fliack

Sources

Built-in Sharing Behavior

https://help.salesforce.com/articleView?id=sharing_across_objects.htm&type=5

Implicit Sharing

https://developer.salesforce.com/docs/atlas.en-us.draes.meta/draes/draes_object_relationships_implicit_sharing.htm

Session 2

Salesforce - Record Locks & Skews

by Nora Nicklis

Record Locks

Types of Record Locks

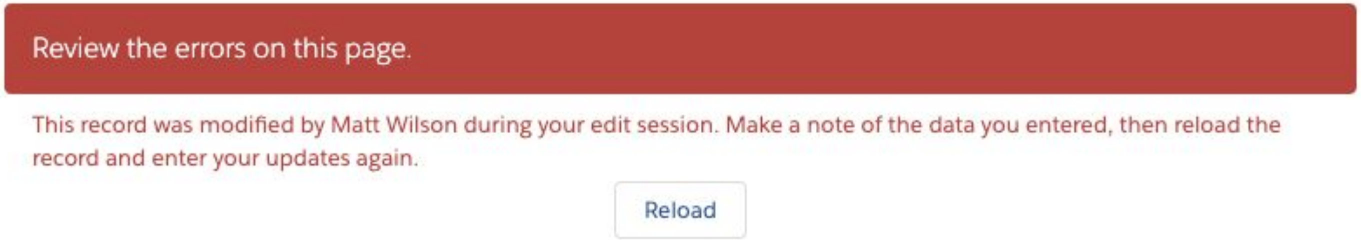
Record Locks can happen

- business-related
  ➔ defined by business process and implemented **intentionally** (e.g. approval process, read-only record types)

- system-related
  ➔ done by Salesforce, available and implemented out of the box (cannot be changed)

Error

UNABLE_TO_LOCK_ROW, unable to obtain exclusive access to this record



Review the errors on this page.

This record was modified by Matt Wilson during your edit session. Make a note of the data you entered, then reload the record and enter your updates again.

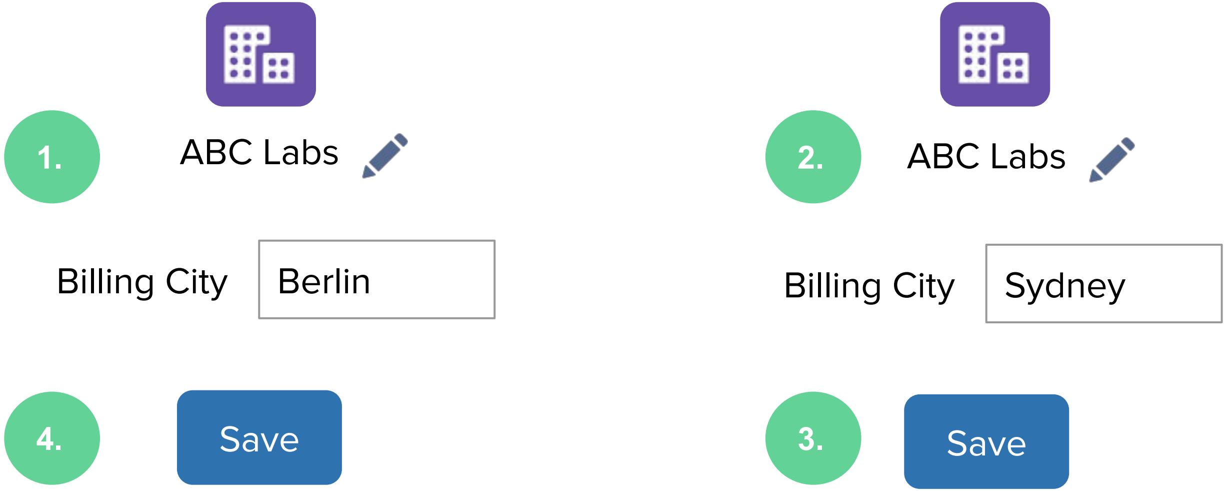
System Related Record Locks

Quality of a database is often measured by its **data integrity:**

- Data accuracy
- Data completeness
- Data consistency

One measure by Salesforce (and many other databases) to maintain data integrity is to place **record locks** on records (and their parent records) while they are being edited.

System Related Record Locks - Example (II)



1. ABC Labs

Billing City: Berlin

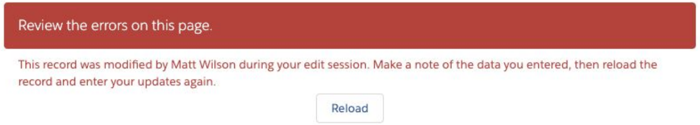
4. Save

2. ABC Labs

Billing City: Sydney

3. Save

5.



Review the errors on this page.

This record was modified by Matt Wilson during your edit session. Make a note of the data you entered, then reload the record and enter your updates again.

System Related Record Locks - Example (I)

Data Load

ID	Account	Rating
0013z00002Q4zFbAAJ	ABC Labs	A

Integration

ID	Account	Rating
0013z00002Q4zFbAAJ	ABC Labs	B

Automation / Apex

ID	Account	Rating
0013z00002Q4zFbAAJ	ABC Labs	C

❗ A given transaction can only **wait a maximum of 10 seconds** for a lock to be released, otherwise it will throw and UNABLE_TO_LOCK_ROW exception

Skews

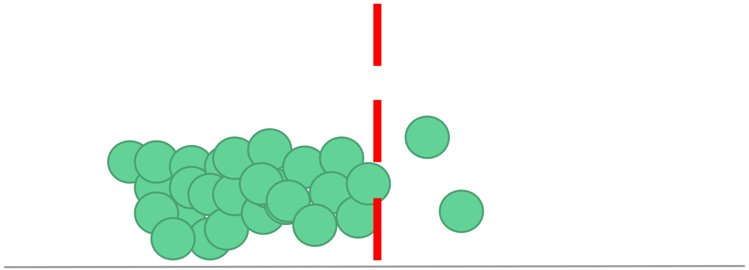
skew

ADJECTIVE

Neither parallel nor at right angles to a specified or implied line; askew; crooked.

Data Skews

In a database, a skew refers to a situation in which there is an **uneven distribution of data**.

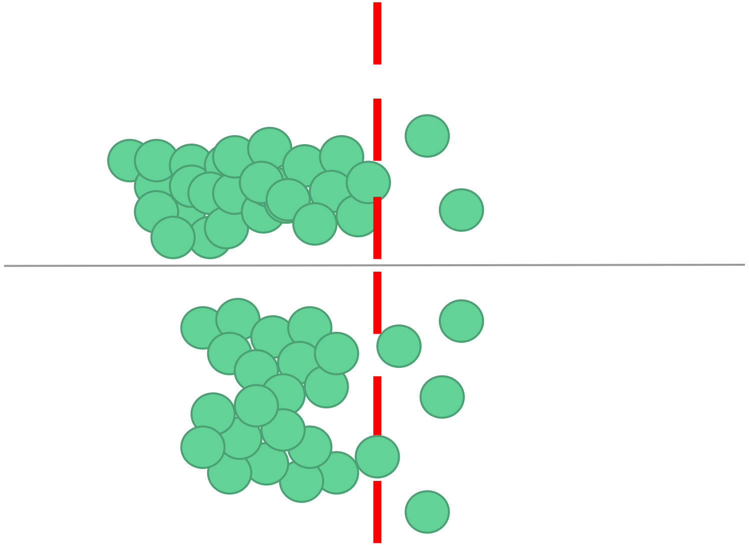


Data Skews

In a database, a skew refers to a situation in which there is an **uneven distribution of data**.

Data Skews

In a database, a skew refers to a situation in which there is an **uneven distribution of data**.



Data Skews in Salesforce

In Salesforce, data skews happen when records relate to one another and **child records are unevenly distributed across the parent records.** They will typically result in **performance issues.**

There are three kinds of data skews that can occur:

- Account Skew
- Ownership Skew
- Lookup Skew

Lookup Skew - Overview

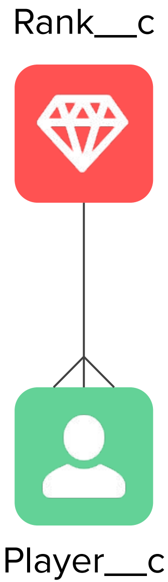
Lookup Skews happen in object relationships when **a large number of child records are associated to one parent record**.

When a child record is inserted or updated, Salesforce locks the parent record the child will be associated to in order to maintain data integrity. When you work with large data, and there are processes (e.g. custom code, automated processes, integrations) you can easily run into lock contention.

❗ Lookup skews happen in lookup relationships AND master-detail relationships
(Locks only occur on lookup relationship if it is not configured to clear the value of this field if the lookup record is deleted)

❗ Lookup skews can effect standard AND custom objects

Lookup Skew - Example



- Scout
- Explorer
- Mountaineer

1.000.000 Players, each will be assigned a rank

Result

➔ Each rank will have approx. 333.333 child player records

➔ Potential lock issues when loading/updating player information

How to Identify a Lookup Skew

There is no exact identifier by which you will be able to identify a lookup skew. Don't wait until somebody complains, try to address to problem right away.

Take a look at the following.

- Objects with a lot of records
- Objects where a lot of inserts and updates happening in parallel
- Objects where you get complaints about locks from users

How to Handle Lookup Skews

Whenever possible, try to avoid Lookup Skews!

If it is not possible to avoid an Lookup Skew, look at the following strategies to soothe them:

- Reducing Record Save Time (Apex Best Practices, Asynchronous processing, Consolidate Automation)
- Picklist fields
- Distributing the skew (Add records on parent, Default = Blank, Review need,)
- Reducing the load (Serial loading, Order by parent ID and chunk the data carefully)

Account Skew - Overview

An Account Skew arises, when **a large number of child records (>10.000) are related to a single account.**

There are two main problems that can be caused by an Account Skew:

- Record Locking
- Sharing Problems

Account Skew - Sharing Problems

Sharing problems arise because of Salesforce built in **parent implicit sharing.**

(Users with access to to contacts, cases and opportunities get read access to the related account)

If sharing of a child record of a certain object is updated, Salesforce will have to check all other children of that object to check, if the implicit share for the Account can be removed.

- Depending on the amount of children, the recalculation can take long and will cause performance issues
- During the whole operation, a lock is placed on the account preventing other from editing it

Account Skew - Example

Dummy Account

Owner: Matt

1.
Lara Smith

Owner: Nora

2.
Joey Young

Owner: Sarah

…

Owner: …

23 456.
Carlos Rodriguez

Owner: Yin

23 457.
Fumiko Suzuki

Owner: Nora

Account Skew - Example

Dummy Account

Owner: Matt

1.
Lara Smith

Owner: Camilla

2.
Joey Young

Owner: Sarah

...

...

Owner: ...

23 456.
Carlos Rodriguez

Owner: Yin

23 457.
Fumiko Suzuki

Owner: Nora

How to Handle Account Skews

Whenever possible, try to avoid Account Skews!

To do so, try to follow these suggestions:

- Limit the amount of Account children per object to 10.000.
- If you need generic Accounts, create a pool of accounts and use a round robin algorithm to distribute them
- Consider setting the Account OWD to “Public Read/Write” (the record locking will stay, but sharing calculations won’t happen)
- Redistribute existing child object records during off hours (Batch Apex, Bulk API)

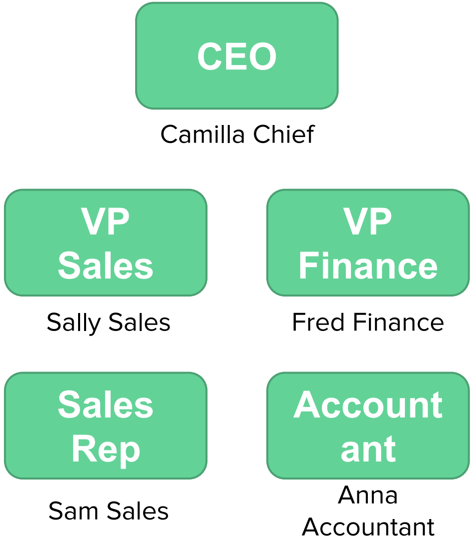
Ownership Skew - Overview

An ownership skew happens, if **a single user owns more than 10.000 records of one object.**

Performance issues caused by an ownership skew will most typically appear in large organisation where **data changes that require sharing recalculation** happen frequently. These changes could happen on:

- Ownership based sharing rules
- Record ownerships
- Group members / Role Hierarchy

Ownership Skew - Example



CEO

Camilla Chief

VP Sales

Sally Sales

VP Finance

Fred Finance

Sales Rep

Sam Sales

Accountant

Anna Accountant

Joey Young

Owner: Sam Sales

Contacts Table

Contact ID	Owner	Name
0033z00002fpwsYAAQ	Sam Sales	Joey Young

Contact Sharing Table

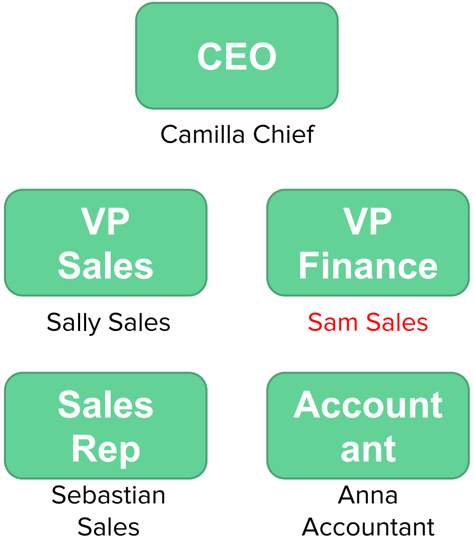
Contact ID	User/Group	Row Cause
0033z00002fpwsYAAQ	Sam Sales	Owner

Group Maintenance Table

User	Groups
Camilla Chief *	CEO
Sally Sales *	VP Sales
Sam Sales	Sales Rep

* implicit sharing

Ownership Skew - Example



CEO
Camilla Chief

VP Sales
Sally Sales

VP Finance
Sam Sales

Sales Rep
Sebastian Sales

Accountant
Anna Accountant

Joey Young

Owner: Sam Sales

Contacts Table

Contact ID	Owner	Name
0033z00002fpwsYAAQ	Sam Sales	Joey Young

Contact Sharing Table

Contact ID	User/Group	Row Cause
0033z00002fpwsYAAQ	Sam Sales	Owner

Group Maintenance Table

User	Groups
Camilla Chief *	CEO
~~Sally Sales~~ *	~~VP Sales~~
Sam Sales	VP Finance

* implicit sharing

Reasons for Ownership Skews

Common reasons for ownership skews:

- The System Administrator is a fallback/default owner during automation processes/data migration
- Usage of a dummy user (for unassigned records, for records created through integrations)

How to Handle Ownership Skews

Whenever possible, try to avoid Ownership Skews!

If it is not possible to avoid an Owernship Skew, try to follow these suggestions:

- Ensure that the “skewed” owner is not part of the role hierarchy or put them in a role at the top of the hierarchy/within its own branch
- Don’t move the user into another role / move the role within the role hierarchy
- Keep the user out of public groups used for sharing
- For organization wide sharing, review OWDs (is private necessary?)
- Consider granting access on profile level or using criteria based sharing rules for sharing with a group of users

Summary

Whenever possible, try to avoid Skews!

If not, don’t worry!

Identify them early and apply strategies learned to soothe/handle them.

Sources

- https://developer.salesforce.com/blogs/engineering/2013/04/managing-lookup-skew-to-avoid-record-lock-exceptions.html
- https://developer.salesforce.com/blogs/engineering/2012/04/avoid-account-data-skew-for-peak-performance.html
- https://resources.docs.salesforce.com/sfdc/pdf/salesforce_record_access_under_the_hood.pdf
- https://developer.salesforce.com/blogs/engineering/2012/06/architect-salesforce-record-ownership-skew-for-peak-performance-in-large-data-volume-environments.html

Summary

- Slides & Recording will be shared after the meeting (Slides → Linkedin, Recording → Youtube)
- Unanswered questions will be answered in the upcoming days in the comments of the video
- We will keep you posted on the topics and the date for the next session
  (if your are interested in being a presenter next time, please reach out)