CHEAT SHEET · TABLE PERMISSIONS

Power Pages Table Permissions Explained

Secure your portal data effectively.

Table permissions are the foundation of data security in Microsoft Power Pages. They control which Dataverse records your portal users can read, create, update, or delete. Without proper table permissions, all data access is blocked by default.

This cheat sheet covers **15 proven security patterns** that solve real-world access control challenges. Whether you're building a customer portal, partner portal, or employee self-service site, these patterns will help you implement secure and scalable data access.

Global Scope

Access all records regardless of ownership. Use sparingly.

Contact Scope

Access records linked to the user's contact record.

Account Scope

Access records linked to the user's parent account.

Self Scope

Access only the user's own contact record.

Parent Scope

Inherit access through a parent table relationship.

Tino Rabe · former Microsoft MVP for Power Pages (2024 to 2026)

15 Power Pages Table Permission Patterns

1

Global Read-Only

Allow all authenticated users to read all records. No write access.

EXAMPLE:

Access: Global
Privileges: Read

2

Contact Scope CRUD

Users can create/read/update/delete only records related to their contact.

EXAMPLE:

Access: Contact
Privileges: Read, Write, Create, Delete

3

Account Scope Read

Users see records associated to their account (organization-wide visibility).

EXAMPLE:

Access: Account
Privileges: Read

4

Self-Only Access

Users can only read/update their own Contact record. Most restrictive.

EXAMPLE:

Access: Self
Privileges: Read, Write

5

Parent-Child Relationship

Access child records via parent table permission. Clean & scalable pattern.

EXAMPLE:

Parent: Case (Contact)
Child: Note (via regardingobjectid)

6

Anonymous Public Access

Allow unauthenticated users to read public records. Use with caution!

EXAMPLE:

Web Role: Anonymous
Access: Global, Read only

7

Create-Only Permission

Users can create records but cannot read them back. Useful for submissions.

EXAMPLE:

Access: Global
Privileges: Create only

8

Read + Append Pattern

Allow users to read records and append related records (e.g., add notes to cases).

EXAMPLE:

Privileges: Read, Append
Parent: Case → Child: Note

9

Multi-Level Parent Scope

Chain permissions across 3+ levels (Contact → Case → Note → Attachment).

EXAMPLE:

Contact → Case
Case → Note → Attachment

10

Account Hierarchy Access

Enable hierarchy security to grant access to parent/child accounts.

EXAMPLE:

Parent Account permissions cascade to child accounts

11

Role-Based Read/Write Split

Different web roles get different privileges (Admins: CRUD, Users: Read).

EXAMPLE:

Admin Role: Read, Write, Delete
User Role: Read only

12

Append To Relationship

Allow linking records to parent records (opposite of Append).

EXAMPLE:

Privilege: AppendTo
Link Note to existing Case

13

No Delete Policy

Grant Read/Write/Create but never Delete. Preserve data integrity.

EXAMPLE:

Privileges: Read, Write, Create
(Delete intentionally omitted)

14

Read-Only for Anonymous, CRUD for Authenticated

Public can browse, logged-in users can interact fully.

EXAMPLE:

Anonymous: Read
Authenticated: Read, Write, Create

15

Polymorphic Lookup Workaround

Polymorphic lookups not supported. Use separate permissions per entity type.

EXAMPLE:

Instead of regardingobjectid,
use case-specific lookups

When to Use Which Table Permission Pattern

Choosing the right table permission pattern depends on your business requirements. Here's a quick decision guide:

For Customer Portals

- → **Contact Scope CRUD** for customer-owned records (cases, orders, invoices)
- → **Self-Only Access** for profile management
- → **Parent-Child Relationship** for related records (case notes, order lines)

For Partner Portals

- → **Account Scope** for organization-wide visibility
- → **Account Hierarchy Access** for multi-level partner structures
- → **Role-Based Split** for partner admins vs. regular users

For Public Websites

- → **Anonymous Public Access** for knowledge articles, FAQs

- → **Create-Only Permission** for contact forms, lead capture
- → **Read-Only Anonymous + CRUD Authenticated** for product catalogs with user reviews

Best Practices

- → Always start with the most restrictive permission and expand as needed
- → Use Parent-Child relationships instead of Global scope whenever possible
- → Test permissions with different user accounts before going live
- → Document your permission structure for future maintenance
- → Consider using Web API column permissions for sensitive fields

Need help implementing table permissions in your Power Pages project? **Book a free consultation** with a former Microsoft MVP (2024 to 2026).

PowerPortals.de
Microsoft Power Pages consulting by **Tino Rabe**, former Microsoft MVP for Power Pages