Comparison: BOI FreeDa vs. MS Access

Criteria	BOI FreeDa	Microsoft Access
Time-to-value	Can be introduced within a few weeks; no definition of masks required	Initial input masks and applications can be created quickly, but they lack resilience as complexity increases
Technology	Web application with centralized access, accessible at any time via a browser	Desktop database (local or shared on the network)
Multi-user operation	Proven architecture for high user volumes (100+ concurrent users)	Limited functionality; ideal for individual users or small teams (up to 10 concurrent users); performance issues with many concurrent users
Scalability & performance	Stable architecture for small to large datasets: from 1 to 10,000+ tables	Suitable for small amounts of data; unstable with larger data loads
Audit trail & compliance	Out of the box: audit trail, versioning, traceable changes including user and timestamps	No built-in versioning or audit trails; audit-proof extensions must be programmed manually
Governance & roles	Fine-grained role and rights management	Limited user and rights management; no granular role-based control
Maintenance processes	Standardized 2-, 4-, and 6-eye processes, including change view	Custom forms; validation rules and approvals must be implemented manually
Database connection	Supports all major relational databases	MS Access databases are often island solutions; integration with other RDBMS is limited
APIs & integration	Standardized interfaces, such as REST APIs, for cross-platform automated processes	Interfaces can only be used to a limited extent and require additional effort; designed for desktop applications

Comparison: BOI FreeDa vs. MS Access

Business operation	On-premises and cloud-ready; Docker-based	local or shared as a file; no professional lifecycle management; no centralized maintenance
Flexibility	High configurability, extensibility via APIs, and standardized modules	Maximum flexibility for small-scale scenarios, but no sustainable governance or scalability
Integration	Standardized APIs and interfaces for integration into business processes	Integration with other systems is complex; typically done via ODBC or manual processes
Usability	Modern web interface designed specifically for table maintenance without requiring database knowledge; no maintenance masks required	Familiar MS Office interface, easy input mask creation, but limited usability and difficult in meeting modern requirements (e.g. accessibility)
Security	Company-wide security mechanisms, logging, and access control	Local file or network security; no continouing security strategies