



ITTIA DB SQL™

Key Features

- Small footprint (min. 150KB)
- Full ACID transactions for recovery and rollback
- Embed in an application or run as a separate process
- No DBA
- Fast indexed search and sort (B+ tree, T-tree)
- Stores data in platform-independent format
- In-memory, on-disk, hybrid
- Direct table/index API
- Runtime SQL API
- Multiple thread and process concurrent access
- Open database files directly
- TCP/IP remote access
- Row-level or storage-level locking
- Supports any processor and operating system
- Supports no OS
- Automatic type conversion
- Encryption callbacks
- Strongly-typed tables
- Strict, group, and lazy completion
- Memory allocation guarantees
- Replication, synchronization, backup
- Two-phase commit

<http://www.ittia.com>
info@ittia.com

ITTIA DB SQL is a lightweight embedded database for embedded systems, intelligent devices, and mobile devices. As a small-footprint database library, ITTIA DB SQL offers developers advanced data management capabilities, including transactions, scalable indexing, shared access, and runtime SQL queries.

Relational Model

As a relational database, ITTIA DB SQL stores data in a straightforward table format. Relationships follow naturally from the data itself, enabling efficient access from a variety of contexts, accelerating development and minimizing maintenance. SQL and the relational model are widely accepted standards for database storage that industry professionals depend on.

Flexibility

Each embedded application has a unique balance of memory footprint, latency, storage, and data sharing requirements. ITTIA DB SQL satisfies a wide range of expectations with a common data management framework and APIs. Select Compact, Standard, or Plus edition, and on-disk, in-memory, or hybrid storage.

ITTIA DB Compact

- Minimum footprint
- Embedded library

ITTIA DB SQL Standard

- Runtime SQL
- Single-user access

ITTIA DB SQL Plus

- Multi-threading
- Client/server

Reliability

ITTIA DB SQL protects data by grouping related changes into atomic transactions. Transaction logging ensures that important information is never lost and the database is protected from corruption. When an unexpected power failure occurs, both in-memory and on-disk tables are automatically repaired.

Scalability

ITTIA DB SQL scales to fully utilize any hardware environment. Robust indexing and logging algorithms guarantee consistent performance for on-disk tables and optimal performance for in-memory tables. ITTIA DB SQL uses main memory intelligently to minimize expensive flash and hard disk operations.

Performance

ITTIA DB SQL is designed for high-performance data management by minimizing overhead and effectively utilizing scarce resources. Applications have the opportunity to access tables and indexes directly, eliminating unnecessary overhead in simple queries. Whether an application needs great overall performance for high-throughput on-disk tables, or low latency access to in-memory tables, ITTIA DB SQL will deliver.

Specifications

Interfaces

- C API
- C++ API
- ODBC driver
- ADO.NET data source
- LuaSQL module
- TCP/IP
- Java API
- .NET API
- PHP module
- Python module
- Ruby Sequel adapter
- Shared memory

SQL

- Runtime DDL: CREATE TABLE, ALTER TABLE, DROP TABLE, CREATE INDEX, etc.
- Transaction isolation: READ COMMITTED, REPEATABLE READ, SERIALIZABLE
- All join types: CROSS, INNER, LEFT OUTER, RIGHT OUTER, FULL OUTER
- JOIN ON, JOIN USING, NATURAL JOIN
- Set operations: UNION, EXCEPT, INTERSECT
- CASE (simple and searched)
- Natural numbers virtual table: \$NAT
- Sequence generators: NEXT VALUE FOR

Data Types

- int, integer, tinyint, smallint, bigint, unsigned
- float, float64, currency
- varchar, nvarchar, utf8str, utf32str, blob
- date, time, datetime, timestamp
- Encodings: Unicode UTF-8, UTF-16, UTF-32, ANSI

Data management methods

- On-disk storage: persistent storage to any block device, such as flash memory, with efficient paging. Uses file I/O or custom block interface.
- In-memory storage: transaction management and indexing for data stored primarily in RAM.
- Hybrid storage: combine on-disk and in-memory tables in a single database.

Data integrity

Transaction rollback and crash recovery compliant with industry-standard ACID properties. Transaction isolation uses either storage-level or row-level locking.

High Availability and Mirroring

- Hot online backup
- Log-based asynchronous two-way replication
- Synchronous replication on commit
- Synchronization with back-end database servers

Target Platforms

- ARM, x86, x64, PowerPC, MIPS, SuperH
- Microsoft® Windows Mobile/CE/PocketPC
- MontaVista® Linux, Debian GNU/Linux, RedHat Linux, WindRiver Linux, MeeGo
- QNX® Neutrino® RTOS
- Express Logic® ThreadX™ and FileX™
- Microsoft® Windows™ 2000/XP/Vista
- Easily ported to virtually any desired platform

Development Platforms

- x86 Microsoft® Windows™ 2000/XP/Vista/7
- x86 Linux

Development Environments

- GNU toolchain
- Microsoft® Visual Studio™
- IAR Embedded Workbench

Database Specifications

Maximum database size:	127 terabytes
Maximum rows per table:	4,294,967,295
Maximum columns per table:	4,294,967,295
Maximum indexes per table:	4,294,967,295
Maximum page size:	32,768 bytes
Maximum row size:	16,363 bytes
Maximum field size:	16,363 bytes
Default maximum row size:	2,027 bytes
Maximum BLOB size:	4,294,967,295 bytes
Maximum open databases:	unlimited
Maximum connections:	unlimited