Structured Query Language (SQL)

Go up to Terms and Definitions

Discussion

Structured Query Language (SQL) provides a standard language for defining, querying, manipulating, and controlling data in relational database systems.

SQL supports database schemas, tables, columns, datatypes, constraints, queries, views, transactions, access control, and stored logic. In architecture work, SQL artifacts express semantic content when table names, column names, constraints, relationships, views, and queries preserve traceability to defined domain meaning.

SQL differs from schema. A schema defines information elements, permitted values, relationships, and constraints within a defined representation context. SQL provides one language for defining and using database schemas in relational database environments.

SQL also differs from Semantic. SQL syntax and database structure do not establish domain meaning on their own. SQL expresses selected semantic content when SQL artifacts trace to a conceptual model, vocabulary, type system, rule set, report model, or other authoritative semantic source.

SQL semantic content includes:

- Table names
- Column names
- Datatypes
- Primary keys
- Foreign keys
- Unique constraints
- Check constraints
- Referential constraints
- Views
- Queries
- Stored procedures
- Triggers
- Transactions
- Access controls
- Audit tables
- Traceability to domain meaning

Definition

standard language for defining, querying, manipulating, and controlling data in relational database systems

Source

Generalised from ISO/IEC 9075 SQL usage, relational database practice, data modeling practice, and software architecture usage; specialized for use in the FX Demo Reference Architecture.

Note

SQL is not equivalent to a conceptual model, schema, ontology, or semantic model. SQL artifacts express selected semantic commitments in a relational database context when their structures, constraints, and queries preserve traceability to governed domain meaning.

A database column can have a valid SQL datatype while still lacking clear domain meaning. For example, a column with datatype `DATE` does not by itself distinguish `tradeDate`, `settlementDate`, `effectiveDate`, or `terminationDate`.

Example

An FX trade table defines selected relational structures for trade information.

SQL artifact	Example	Semantic interpretation
Table	`FX_TRADE`	relational representation of FX trade records
Column	`TRADE_IDENTIFIER`	identifier for the trade
Column	`COUNTERPARTY_LEI`	legal entity identifier for a counterparty
Column	`CURRENCY_PAIR`	ordered pair of currencies exchanged in the trade
Column	`NOTIONAL_AMOUNT`	amount used to calculate settlement obligations
Column	`TRADE_DATE`	date on which the trade occurs
Column	`SETTLEMENT_DATE`	date on which settlement occurs
Check constraint	`SETTLEMENT_DATE >= TRADE_DATE`	relational enforcement of the lifecycle rule that settlement date does not precede trade date

The SQL definition constrains the relational representation. The semantic meaning comes from the governed vocabulary, conceptual model, type system, lifecycle rules, and validation rules that explain what each table, column, and constraint means.

```
CREATE TABLE FX_TRADE (
    TRADE_IDENTIFIER VARCHAR(64) PRIMARY KEY,
    COUNTERPARTY_LEI  VARCHAR(20) NOT NULL,
    CURRENCY_PAIR     VARCHAR(7) NOT NULL,
    NOTIONAL_AMOUNT   DECIMAL(18,2) NOT NULL,
    TRADE_DATE        DATE NOT NULL,
    SETTLEMENT_DATE   DATE NOT NULL,
    LIFECYCLE_STATE   VARCHAR(32) NOT NULL,
```

```
    CHECK (SETTLEMENT_DATE >= TRADE_DATE)
);
```

The CHECK constraint expresses rule semantics in SQL form. The constraint has governed meaning only when it traces back to the FX trade lifecycle rule that defines a valid relationship between the trade date and the settlement date.
