**Table of Contents**

1. Overview
2. Database Creation
3. Schema Creation
4. Table Creation
5. Data Types
6. Constraints
7. Best Practices
8. Examples
9. Troubleshooting

**Overview**

This documentation covers Data Definition Language (DDL) commands for creating and managing databases, schemas, and tables in Snowflake. Snowflake's DDL syntax follows SQL standards with some Snowflake-specific extensions.

**Database Creation**

**Basic Database Syntax**

CREATE DATABASE [IF NOT EXISTS] <database\_name>

[CLONE <source\_database>]

[DATA\_RETENTION\_TIME\_IN\_DAYS = <days>]

[COMMENT = '<comment>']

[TAG (<tag\_name> = '<tag\_value>'[, ...])];

**Examples**

sql

-- Simple database

CREATE DATABASE sales\_db;

-- Database with comment and retention

CREATE DATABASE analytics\_db

DATA\_RETENTION\_TIME\_IN\_DAYS = 90

COMMENT = 'Database for analytical processing';

-- Clone existing database

CREATE DATABASE sales\_db\_dev CLONE sales\_db;

-- Transient database

CREATE TRANSIENT DATABASE staging\_db

DATA\_RETENTION\_TIME\_IN\_DAYS = 1;

**Database Properties**

| Property | Description | Default | Range |
| --- | --- | --- | --- |
| DATA\_RETENTION\_TIME\_IN\_DAYS | Time Travel retention period | 1 | 0-90 |
| TRANSIENT | No Fail-safe protection | FALSE | Boolean |
| COMMENT | Descriptive comment | NULL | String |

**Schema Creation**

**Basic Schema Syntax**

sql

CREATE SCHEMA [IF NOT EXISTS] <schema\_name>

[DATA\_RETENTION\_TIME\_IN\_DAYS = <days>]

[COMMENT = '<comment>']

[WITH MANAGED ACCESS]

[TAG (<tag\_name> = '<tag\_value>'[, ...])];

**Examples**

sql

-- Simple schema

CREATE SCHEMA raw\_data;

-- Schema with retention and comment

CREATE SCHEMA processed\_data

DATA\_RETENTION\_TIME\_IN\_DAYS = 30

COMMENT = 'Schema for processed data';

-- Managed access schema

CREATE SCHEMA secure\_data

WITH MANAGED ACCESS;

-- Transient schema

CREATE TRANSIENT SCHEMA temp\_data

DATA\_RETENTION\_TIME\_IN\_DAYS = 1;

**Schema Types**

| Type | Description | Use Case |
| --- | --- | --- |
| Standard | Regular schema with full features | Most use cases |
| Transient | No Fail-safe, shorter retention | Temporary data |
| Managed Access | Centralized privilege management | Security-sensitive data |

**Table Creation**

**Basic Table Syntax**

sql

CREATE [TRANSIENT] TABLE <table\_name> (

<column\_name> <data\_type> [DEFAULT <expression>] [AUTOINCREMENT] [NOT NULL] [COMMENT '<comment>'],

...

[CONSTRAINT <constraint\_name> PRIMARY KEY (<column\_list>)]

[CONSTRAINT <constraint\_name> UNIQUE (<column\_list>)]

[CONSTRAINT <constraint\_name> FOREIGN KEY (<column\_list>) REFERENCES <ref\_table>(<ref\_columns>)]

)

[CLUSTER BY (<column\_list>)]

[DATA\_RETENTION\_TIME\_IN\_DAYS = <days>]

[COMMENT = '<comment>']

[TAG (<tag\_name> = '<tag\_value>'[, ...])];

**Examples**

sql

-- Simple table

CREATE TABLE customers (

customer\_id NUMBER AUTOINCREMENT,

first\_name VARCHAR(50),

last\_name VARCHAR(100),

email VARCHAR(255),

created\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP()

);

-- Table with constraints

CREATE TABLE orders (

order\_id NUMBER AUTOINCREMENT PRIMARY KEY,

customer\_id NUMBER NOT NULL,

order\_date DATE NOT NULL,

total\_amount NUMBER(10,2) CHECK (total\_amount >= 0),

status VARCHAR(20) DEFAULT 'PENDING',

CONSTRAINT fk\_customer FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id)

);

-- Table with clustering

CREATE TABLE sales\_transactions (

transaction\_id NUMBER AUTOINCREMENT,

sale\_date DATE,

region VARCHAR(50),

amount NUMBER(10,2),

CLUSTER BY (sale\_date, region)

);

-- Transient table

CREATE TRANSIENT TABLE temp\_sessions (

session\_id VARCHAR(100),

user\_data VARIANT

) DATA\_RETENTION\_TIME\_IN\_DAYS = 1;

**Data Types**

**Common Snowflake Data Types**

| Category | Data Types | Description |
| --- | --- | --- |
| Numeric | NUMBER, DECIMAL, INT, FLOAT | Precision and scale configurable |
| String | VARCHAR, CHAR, STRING | Variable-length character data |
| Binary | BINARY, VARBINARY | Binary data |
| Boolean | BOOLEAN | TRUE/FALSE values |
| Date/Time | DATE, TIME, TIMESTAMP, TIMESTAMP\_LTZ | Date and time values |
| Semi-structured | VARIANT, ARRAY, OBJECT | JSON-like data |
| Geospatial | GEOGRAPHY | Spatial data |

**Data Type Examples**

sql

CREATE TABLE data\_types\_demo (

-- Numeric types

id NUMBER AUTOINCREMENT,

price DECIMAL(10,2),

quantity INTEGER,

-- String types

name VARCHAR(100),

description STRING,

-- Date/Time types

created\_date DATE,

created\_ts TIMESTAMP DEFAULT CURRENT\_TIMESTAMP(),

-- Boolean

is\_active BOOLEAN DEFAULT TRUE,

-- Semi-structured

metadata VARIANT,

tags ARRAY,

-- Geospatial

location GEOGRAPHY

);

**Constraints**

**Supported Constraints**

| Constraint | Description | Enforcement |
| --- | --- | --- |
| PRIMARY KEY | Uniquely identifies records | Not enforced |
| UNIQUE | Ensures column uniqueness | Not enforced |
| FOREIGN KEY | Enforces referential integrity | Not enforced |
| NOT NULL | Prevents NULL values | Enforced |
| CHECK | Custom validation conditions | Enforced |

**Constraint Examples**

sql

CREATE TABLE constrained\_table (

id NUMBER AUTOINCREMENT PRIMARY KEY,

user\_id NUMBER NOT NULL,

email VARCHAR(255) UNIQUE NOT NULL,

age INTEGER CHECK (age >= 0),

created\_date DATE DEFAULT CURRENT\_DATE(),

-- Composite primary key

CONSTRAINT pk\_composite PRIMARY KEY (id, user\_id),

-- Foreign key

CONSTRAINT fk\_user FOREIGN KEY (user\_id) REFERENCES users(id)

);

**Best Practices**

**Naming Conventions**

sql

-- Databases: lowercase with underscores

CREATE DATABASE sales\_analytics;

-- Schemas: descriptive, layered approach

CREATE SCHEMA raw;

CREATE SCHEMA staging;

CREATE SCHEMA mart;

-- Tables: plural, descriptive names

CREATE TABLE customers;

CREATE TABLE sales\_orders;

CREATE TABLE product\_categories;

-- Columns: snake\_case

customer\_id, first\_name, created\_date

**Performance Optimization**

sql

-- Use clustering keys for large tables

CREATE TABLE large\_table (

id NUMBER,

date\_column DATE,

region VARCHAR(50),

CLUSTER BY (date\_column, region)

);

-- Appropriate data types

CREATE TABLE optimized\_table (

-- Use smallest practical numeric precision

small\_id INTEGER, -- Instead of NUMBER(38,0)

price NUMBER(10,2), -- Appropriate precision

-- Use VARCHAR instead of STRING for defined lengths

product\_code VARCHAR(20),

-- Use DATE instead of TIMESTAMP when time not needed

event\_date DATE

);

**Data Management**

sql

-- Set appropriate retention

CREATE TRANSIENT TABLE temp\_data

DATA\_RETENTION\_TIME\_IN\_DAYS = 1;

-- Use comments for documentation

CREATE TABLE customers (

customer\_id NUMBER COMMENT 'Surrogate key for customer',

email VARCHAR(255) COMMENT 'Customer email address, must be unique'

) COMMENT = 'Master customer data table';

-- Implement soft deletes

CREATE TABLE products (

product\_id NUMBER AUTOINCREMENT,

product\_name VARCHAR(255),

is\_active BOOLEAN DEFAULT TRUE,

deleted\_at TIMESTAMP NULL

);

**Complete Examples**

**E-commerce Database Setup**

sql

-- Create database

CREATE DATABASE ecommerce

DATA\_RETENTION\_TIME\_IN\_DAYS = 90

COMMENT = 'E-commerce platform database';

USE DATABASE ecommerce;

-- Create schemas

CREATE SCHEMA raw

COMMENT = 'Raw data from source systems';

CREATE SCHEMA staging

COMMENT = 'Cleaned and validated data';

CREATE SCHEMA analytics

COMMENT = 'Data for reporting and analysis';

-- Create tables in raw schema

USE SCHEMA raw;

CREATE TABLE customers (

customer\_id NUMBER AUTOINCREMENT PRIMARY KEY,

first\_name VARCHAR(100) NOT NULL,

last\_name VARCHAR(100) NOT NULL,

email VARCHAR(255) UNIQUE NOT NULL,

date\_of\_birth DATE,

registration\_date DATE DEFAULT CURRENT\_DATE(),

country VARCHAR(50),

CLUSTER BY (registration\_date, country)

) COMMENT = 'Customer master data';

CREATE TABLE products (

product\_id NUMBER AUTOINCREMENT PRIMARY KEY,

product\_name VARCHAR(255) NOT NULL,

category VARCHAR(100),

price NUMBER(10,2) CHECK (price >= 0),

cost NUMBER(8,2) CHECK (cost >= 0),

supplier\_id NUMBER,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP(),

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP()

) COMMENT = 'Product catalog';

CREATE TABLE orders (

order\_id NUMBER AUTOINCREMENT PRIMARY KEY,

customer\_id NUMBER NOT NULL,

order\_date TIMESTAMP NOT NULL,

status VARCHAR(20) DEFAULT 'PENDING',

total\_amount NUMBER(10,2) CHECK (total\_amount >= 0),

shipping\_address VARIANT,

CONSTRAINT fk\_order\_customer FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id),

CLUSTER BY (order\_date, status)

) COMMENT = 'Customer orders';

-- Analytics schema tables

USE SCHEMA analytics;

CREATE TABLE daily\_sales (

sale\_date DATE,

product\_id NUMBER,

total\_quantity NUMBER,

total\_revenue NUMBER(15,2),

avg\_price NUMBER(10,2),

CLUSTER BY (sale\_date)

) COMMENT = 'Daily sales aggregations';

**Data Lake Setup**

sql

-- Data lake database

CREATE DATABASE data\_lake

COMMENT = 'Raw data lake for all source data';

USE DATABASE data\_lake;

-- Source-specific schemas

CREATE SCHEMA web\_events

COMMENT = 'Website tracking events';

CREATE SCHEMA mobile\_events

COMMENT = 'Mobile app events';

CREATE SCHEMA crm\_data

COMMENT = 'CRM system data';

-- Event data table with VARIANT for flexibility

USE SCHEMA web\_events;

CREATE TABLE page\_views (

event\_id NUMBER AUTOINCREMENT,

session\_id VARCHAR(100),

user\_id NUMBER,

event\_timestamp TIMESTAMP,

page\_url VARCHAR(500),

event\_data VARIANT, -- Flexible JSON data

referrer VARCHAR(500),

user\_agent VARCHAR(1000),

-- Computed columns for frequently accessed fields

page\_title VARCHAR(200) AS (event\_data:page\_title::VARCHAR),

time\_on\_page NUMBER AS (event\_data:time\_on\_page::NUMBER),

CLUSTER BY (event\_timestamp, user\_id)

) COMMENT = 'Web page view events with full event data';

**Permission Requirements**

| Operation | Required Privileges |
| --- | --- |
| Create Database | CREATE DATABASE |
| Create Schema | CREATE SCHEMA on database |
| Create Table | CREATE TABLE on schema |
| Create Tag | CREATE TAG (if using tags) |