SQL Complete Guide: Data Types and Commands

Trip Application Database Examples

SQL Data Types

Numeric Data Types

- **INT/INTEGER**: Whole numbers (-2,147,483,648 to 2,147,483,647)
- **BIGINT**: Large whole numbers
- **DECIMAL(p,s)**: Fixed-point numbers with precision and scale
- **FLOAT**: Approximate floating-point numbers
- **DOUBLE**: Double-precision floating-point numbers

String Data Types

- **VARCHAR(n)**: Variable-length character string (max n characters)
- **CHAR(n)**: Fixed-length character string (exactly n characters)
- **TEXT**: Large variable-length text
- **LONGTEXT**: Very large text data

Date and Time Data Types

- **DATE**: Date only (YYYY-MM-DD)
- **TIME**: Time only (HH:MM:SS)
- **DATETIME**: Date and time combined
- **TIMESTAMP**: Date and time with timezone awareness

Other Data Types

- **BOOLEAN**: True/False values
- **BLOB**: Binary Large Objects (images, files)
- **JSON**: JSON formatted data (MySQL 5.7+)

DDL (Data Definition Language)

Commands that define and modify database structure

Key DDL Commands:

- **CREATE**: Creates database objects

- **ALTER**: Modifies existing database objects
- **DROP**: Deletes database objects
- **TRUNCATE**: Removes all data from table (keeps structure)

Example - Creating Trip Application Tables:

```sql
-- Create database
CREATE DATABASE TripApplication;

-- Create Users table
CREATE TABLE users (
    user_id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) NOT NULL UNIQUE,
    email VARCHAR(100) NOT NULL,
    password_hash VARCHAR(255) NOT NULL,
    full_name VARCHAR(100),
    phone VARCHAR(15),
    date_of_birth DATE,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- Create Trips table
CREATE TABLE trips (
    trip_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT,
    destination VARCHAR(100) NOT NULL,
    start_date DATE NOT NULL,
    end_date DATE NOT NULL,
    budget DECIMAL(10,2),
    trip_status VARCHAR(20) DEFAULT 'planned',
    description TEXT,
    FOREIGN KEY (user_id) REFERENCES users(user_id)
);

-- Alter table to add new column
ALTER TABLE trips ADD COLUMN trip_type VARCHAR(30);

-- Drop table
DROP TABLE IF EXISTS temp_bookings;
```

DML (Data Manipulation Language)

Commands that manipulate data within tables

Key DML Commands:

- **INSERT**: Adds new records
- **UPDATE**: Modifies existing records
- **DELETE**: Removes records

Examples - Trip Application Data Operations:

```sql
-- INSERT: Add new users
INSERT INTO users (username, email, password_hash, full_name, phone, date_of_birth)
VALUES
    ('john_traveler', 'john@email.com', 'hashed_password_123', 'John Smith', '+1234567890', '1990-05-15'),
    ('sarah_explorer', 'sarah@email.com', 'hashed_password_456', 'Sarah Johnson', '+1987654321', '1985-12-03');

-- INSERT: Add trip records
INSERT INTO trips (user_id, destination, start_date, end_date, budget, trip_type, description)
VALUES
    (1, 'Paris, France', '2024-06-15', '2024-06-22', 2500.00, 'vacation', 'Romantic getaway to Paris'),
    (2, 'Tokyo, Japan', '2024-08-10', '2024-08-20', 3500.00, 'business', 'Business conference and sightseeing');

-- UPDATE: Modify trip budget
UPDATE trips
SET budget = 3000.00, description = 'Extended romantic getaway to Paris'
WHERE trip_id = 1;

-- UPDATE: Change trip status
UPDATE trips
SET trip_status = 'completed'
WHERE end_date < CURDATE();

-- DELETE: Remove cancelled trip
DELETE FROM trips
WHERE trip_id = 2 AND trip_status = 'cancelled';
```

DQL (Data Query Language)

Commands that retrieve data from database

Key DQL Command:

- **SELECT**: Retrieves data from tables

Examples - Trip Application Queries:

sql

```
-- Basic SELECT: Get all trips
SELECT * FROM trips;

-- SELECT with conditions: Get active trips for a user
SELECT trip_id, destination, start_date, end_date, budget
FROM trips
WHERE user_id = 1 AND trip_status = 'planned';

-- SELECT with JOIN: Get trip details with user information
SELECT
    u.full_name,
    u.email,
    t.destination,
    t.start_date,
    t.end_date,
    t.budget,
    t.trip_status
FROM users u
JOIN trips t ON u.user_id = t.user_id
WHERE t.start_date >= CURDATE();

-- SELECT with aggregation: Get total budget by user
SELECT
    u.full_name,
    COUNT(t.trip_id) as total_trips,
    SUM(t.budget) as total_budget,
    AVG(t.budget) as avg_budget
FROM users u
LEFT JOIN trips t ON u.user_id = t.user_id
GROUP BY u.user_id, u.full_name
HAVING total_trips > 0;

-- SELECT with sorting and limiting
SELECT destination, budget, start_date
FROM trips
WHERE trip_status = 'planned'
ORDER BY start_date ASC, budget DESC
LIMIT 5;
```

DCL (Data Control Language)

Commands that control access and permissions

Key DCL Commands:

- **GRANT**: Gives privileges to users
- **REVOKE**: Removes privileges from users

Examples - Trip Application Security:

```sql
-- Create users with different roles
CREATE USER 'trip_admin'@'localhost' IDENTIFIED BY 'admin_password123';
CREATE USER 'trip_user'@'localhost' IDENTIFIED BY 'user_password456';
CREATE USER 'trip_readonly'@'localhost' IDENTIFIED BY 'readonly_password789';

-- GRANT: Give full access to admin
GRANT ALL PRIVILEGES ON TripApplication.* TO 'trip_admin'@'localhost';

-- GRANT: Give specific permissions to regular user
GRANT SELECT, INSERT, UPDATE ON TripApplication.users TO 'trip_user'@'localhost';
GRANT SELECT, INSERT, UPDATE, DELETE ON TripApplication.trips TO 'trip_user'@'localhost';

-- GRANT: Give read-only access for reporting
GRANT SELECT ON TripApplication.* TO 'trip_readonly'@'localhost';

-- REVOKE: Remove specific permissions
REVOKE DELETE ON TripApplication.trips FROM 'trip_user'@'localhost';

-- REVOKE: Remove all permissions
REVOKE ALL PRIVILEGES ON TripApplication.* FROM 'trip_user'@'localhost';

-- Apply changes
FLUSH PRIVILEGES;
```

Quick Reference Summary

Category	**Commands**	**Purpose**
DDL	CREATE, ALTER, DROP, TRUNCATE	Define/modify database structure
DML	INSERT, UPDATE, DELETE	Manipulate data in tables
DQL	SELECT	Query/retrieve data
DCL	GRANT, REVOKE	Control user access and permissions

Trip Application Database Schema Overview:

```
TripApplication Database
├── users table
│   ├── user_id (INT, Primary Key)
│   ├── username (VARCHAR)
│   ├── email (VARCHAR)
│   ├── password_hash (VARCHAR)
│   ├── full_name (VARCHAR)
│   ├── phone (VARCHAR)
│   ├── date_of_birth (DATE)
│   └── created_at (TIMESTAMP)
│
└── trips table
    ├── trip_id (INT, Primary Key)
    ├── user_id (INT, Foreign Key)
    ├── destination (VARCHAR)
    ├── start_date (DATE)
    ├── end_date (DATE)
    ├── budget (DECIMAL)
    ├── trip_status (VARCHAR)
    ├── trip_type (VARCHAR)
    └── description (TEXT)
```

Study Tips:

1. **Practice data types** by creating tables with different column types
2. **Master JOIN operations** for complex queries combining multiple tables
3. **Understand the difference** between DDL (structure) and DML (data)
4. **Always use proper indexing** on frequently queried columns
5. **Practice with real scenarios** like the trip application examples above