TIME TABLE MANAGEMENT SYSTEM

Complete SQL Documentation

Queries | Database Schema | Sample Outputs

Generated:	15-11-2025 18:35:52
Version:	2.0 (GUI Edition)
Database:	SQLite
Status:	Production Ready

TABLE OF CONTENTS

1. SYSTEM OVERVIEW

The Time Table Management System is a comprehensive solution for managing educational institution schedules. It uses SQLite database to store and manage:

- **Teachers:** Faculty information with departments and specializations
- **Classrooms:** Room details with capacity information
- **Subjects:** Course information with codes and credits
- **Time Slots:** Available class periods throughout the day
- **Time Table:** Mapping of teachers, subjects, and classrooms to specific times

2. DATABASE SCHEMA

The system uses 6 main tables with relationships to maintain data integrity:

Table	Purpose	Key Fields
teachers	Store teacher information	teacher_id, name, department, email
classrooms	Store classroom details	classroom_id, class_name, capacity
subjects	Store subject/course info	subject_id, subject_code, credits
time_slots	Store available time slots	slot_id, start_time, end_time
days	Store days of week	day_id, day_name, day_order
timetable	Main mapping table	classroom_id, teacher_id, subject_id, day_id, slot_id

3. TABLE STRUCTURES & FIELD DEFINITIONS

3.1 Teachers Table

Field	Type	Description
teacher_id	INTEGER	Primary key, auto-increment
teacher_name	TEXT	Name of the teacher (NOT NULL)
department	TEXT	Department name (NOT NULL)
specialization	TEXT	Area of expertise
contact_number	TEXT	Phone number
email	TEXT	Email address (UNIQUE)
created_at	TIMESTAMP	Record creation time

3.2 Classrooms Table

Field	Type	Description
classroom_id	INTEGER	Primary key, auto-increment
class_name	TEXT	Name of classroom (UNIQUE, NOT NULL)
capacity	INTEGER	Maximum students in class
room_number	TEXT	Room number
building	TEXT	Building name
created_at	TIMESTAMP	Record creation time

3.3 Subjects Table

Field	Type	Description
subject_id	INTEGER	Primary key, auto-increment
subject_name	TEXT	Subject name (UNIQUE, NOT NULL)
subject_code	TEXT	Subject code (UNIQUE, NOT NULL)
credits	INTEGER	Course credits
department	TEXT	Department offering subject
created_at	TIMESTAMP	Record creation time

3.4 Time Slots Table

Field	Type	Description
slot_id	INTEGER	Primary key, auto-increment
slot_name	TEXT	Name of slot (UNIQUE, NOT NULL)
start_time	TIME	Class start time
end_time	TIME	Class end time
duration_minutes	INTEGER	Duration in minutes
created_at	TIMESTAMP	Record creation time

3.5 Days Table

Field	Type	Description
day_id	INTEGER	Primary key, auto-increment
day_name	TEXT	Name of day (UNIQUE, NOT NULL)
day_order	INTEGER	Day sequence (UNIQUE)
created_at	TIMESTAMP	Record creation time

3.6 TimeTable (Main) Table

Field	Type	Description
timetable_id	INTEGER	Primary key, auto-increment
classroom_id	INTEGER	FK to classrooms table
teacher_id	INTEGER	FK to teachers table
subject_id	INTEGER	FK to subjects table
day_id	INTEGER	FK to days table
slot_id	INTEGER	FK to time_slots table
academic_year	TEXT	Year (e.g., 2024-2025)
semester	INTEGER	1 or 2
session_type	TEXT	Lecture/Tutorial/Practical/Lab
status	TEXT	Active or Inactive
created_at	TIMESTAMP	Record creation time
updated_at	TIMESTAMP	Last update time

4. SAMPLE DATA

4.1 Sample Teachers

ID	Name	Department	Specialization
1	Dr. John Smith	Computer Science	Database Systems
2	Prof. Sarah Johnson	Mathematics	Calculus
3	Dr. Michael Chen	Physics	Quantum Mechanics
4	Prof. Emma Wilson	English	Literature
5	Dr. Rajesh Kumar	Computer Science	Web Development

4.2 Sample Classrooms

ID	Class Name	Capacity	Room
1	Class A1	50	101
2	Class A2	50	102
3	Class B1	40	201
4	Class B2	40	202
5	Lab 1	30	301

4.3 Sample Subjects

ID	Code	Name	Credits
1	CS301	Database Management Systems	3
2	MATH201	Advanced Mathematics	4
3	PHY301	Quantum Physics	3
4	ENG201	British Literature	3
5	CS305	Web Development	3

4.4 Sample Time Slots

ID	Slot Name	Start	End
1	Slot 1	08:00	09:00
2	Slot 2	09:00	10:00
3	Slot 3	10:15	11:15
4	Slot 4	11:15	12:15
5	Slot 5	13:00	14:00

6	Slot 6	14:00	15:00
7	Slot 7	15:15	16:15
8	Slot 8	16:15	17:15

5. SQL QUERIES & RESULTS

5.1 Query: Select All Teachers

```
SQL Query:
SELECT teacher_id, teacher_name, department, email FROM teachers ORDER BY
teacher_id;
```

Output:

Teacher ID	Name	Department	Email
1	Dr. John Smith	Computer Science	john.smith@university.edu
2	Prof. Sarah Johnson	Mathematics	sarah.johnson@university.edu
3	Dr. Michael Chen	Physics	michael.chen@university.edu
4	Prof. Emma Wilson	English	emma.wilson@university.edu
5	Dr. Rajesh Kumar	Computer Science	rajesh.kumar@university.edu

5.2 Query: Get Time Table with All Details

```
SQL Query:
SELECT t.timetable_id, c.class_name, te.teacher_name, s.subject_code, d.day_name, ts.slot_name,
t.session_type FROM timetable t JOIN classrooms c ON t.classroom_id = c.classroom_id JOIN teachers
te ON t.teacher_id = te.teacher_id JOIN subjects s ON t.subject_id = s.subject_id JOIN days d ON
t.day_id = d.day_id JOIN time_slots ts ON t.slot_id = ts.slot_id ORDER BY d.day_order,
ts.start_time;
```

Output (Sample):

ID	Class	Teacher	Subject	Day	Slot	Type
1	Class A1	Dr. John Smith	CS301	Monday	Slot 1	Lecture
5	Lab 1	Dr. Rajesh Kumar	CS305	Monday	Slot 2	Lab
6	Class A1	Prof. Sarah Johnson	MATH201	Monday	Slot 3	Lecture
2	Class A2	Prof. Sarah Johnson	MATH201	Tuesday	Slot 2	Lecture
3	Class B1	Dr. Michael Chen	PHY301	Wednesday	Slot 3	Lecture
8	Class B1	Dr. John Smith	CS301	Wednesday	Slot 5	Practical
33	Class A2	Prof. Sarah Johnson	MATH201	Wednesday	Slot 5	Tutorial
4	Class B2	Prof. Emma Wilson	ENG201	Thursday	Slot 4	Lecture

5.3 Query: Count Classes by Day

```
SQL Query:
SELECT d.day_name, COUNT(t.timetable_id) as class_count FROM days d LEFT JOIN
timetable t ON d.day_id = t.day_id GROUP BY d.day_id, d.day_name ORDER BY
d.day_order;
```

Output:

Day	Number of Classes
Monday	3
Tuesday	1
Wednesday	3
Thursday	1
Friday	0
Saturday	0

5.4 Query: Get Schedule for Specific Day (Monday)

```
SQL Query:
SELECT ts.start_time, ts.end_time, c.class_name, te.teacher_name, s.subject_code FROM
timetable t JOIN classrooms c ON t.classroom_id = c.classroom_id JOIN teachers te ON
t.teacher_id = te.teacher_id JOIN subjects s ON t.subject_id = s.subject_id JOIN days d
ON t.day_id = d.day_id JOIN time_slots ts ON t.slot_id = ts.slot_id WHERE d.day_name =
'Monday' ORDER BY ts.start_time;
```

Output:

Start	End	Class	Teacher	Subject
08:00	09:00	Class A1	Dr. John Smith	CS301
09:00	10:00	Lab 1	Dr. Rajesh Kumar	CS305
10:15	11:15	Class A1	Prof. Sarah Johnson	MATH201

5.5 Query: Get Schedule for Specific Teacher

```
SQL Query:
SELECT d.day_name, ts.start_time, c.class_name, s.subject_code, t.session_type FROM
timetable t JOIN classrooms c ON t.classroom_id = c.classroom_id JOIN teachers te ON
t.teacher_id = te.teacher_id JOIN subjects s ON t.subject_id = s.subject_id JOIN days d
ON t.day_id = d.day_id JOIN time_slots ts ON t.slot_id = ts.slot_id WHERE
te.teacher_name = 'Dr. John Smith' ORDER BY d.day_order, ts.start_time;
```

Output:

Day	Time	Class	Subject	Type
Monday	08:00	Class A1	CS301	Lecture
Wednesday	13:00	Class B1	CS301	Practical

5.6 INSERT Query Example

```
SQL Query:
INSERT INTO timetable (classroom_id, teacher_id, subject_id, day_id, slot_id,
academic_year, semester, session_type) VALUES (1, 1, 1, 1, 1, '2024-2025', 1,
'Lecture');
```

Description: Inserts a new time table entry linking classroom 1, teacher 1, subject 1 for Monday morning slot.

5.7 UPDATE Query Example

```
SQL Query:
UPDATE timetable SET teacher_id = 2, updated_at = CURRENT_TIMESTAMP WHERE
timetable_id = 1;
```

Description: Updates the teacher assignment for time table entry with ID 1, changing it to teacher 2.

5.8 DELETE Query Example

```
SQL Query:
DELETE FROM timetable WHERE timetable_id = 1;
```

Description: Deletes the time table entry with ID 1.

5.9 Query: System Statistics

```
SQL Query:
SELECT (SELECT COUNT(*) FROM teachers) as total_teachers, (SELECT COUNT(*) FROM
classrooms) as total_classrooms, (SELECT COUNT(*) FROM subjects) as total_subjects,
(SELECT COUNT(*) FROM time_slots) as total_slots, (SELECT COUNT(*) FROM timetable) as
total_entries;
```

Output:

Total Teachers	Total Classrooms	Total Subjects	Total Slots	Total Entries
5	5	5	8	8

6. ADDITIONAL USEFUL QUERIES

6.1 Get classroom schedule

```
SELECT ts.start_time, te.teacher_name, s.subject_code, d.day_name FROM timetable t JOIN
classrooms c ON t.classroom_id = c.classroom_id JOIN teachers te ON t.teacher_id =
te.teacher_id JOIN subjects s ON t.subject_id = s.subject_id JOIN days d ON t.day_id =
d.day_id JOIN time_slots ts ON t.slot_id = ts.slot_id WHERE c.class_name = 'Class A1'
ORDER BY d.day_order, ts.start_time;
```

6.2 Find available time slots

```
SELECT ts.slot_id, ts.slot_name, ts.start_time, ts.end_time FROM time_slots ts WHERE
ts.slot_id NOT IN ( SELECT DISTINCT slot_id FROM timetable WHERE day_id = 1 AND
academic_year = '2024-2025' ) ORDER BY ts.start_time;
```

6.3 Count classes per teacher

```
SELECT te.teacher_name, COUNT(t.timetable_id) as class_count FROM teachers te LEFT JOIN
timetable t ON te.teacher_id = t.teacher_id GROUP BY te.teacher_id, te.teacher_name
ORDER BY class_count DESC;
```

6.4 Get subjects per department

```
SELECT department, COUNT(*) as subject_count FROM subjects GROUP BY department ORDER BY
subject_count DESC;
```

7. SYSTEM SUMMARY

- **Total Tables:** 6 (teachers, classrooms, subjects, time_slots, days, timetable)
- **Database Type:** SQLite (Relational Database)
- **Primary Keys:** All tables have auto-increment primary keys
- **Foreign Keys:** timetable table has 5 foreign key constraints
- **Indexes:** Created on frequently searched fields
- **Data Validation:** NOT NULL, UNIQUE, and DEFAULT constraints applied
- **Sample Data:** Pre-loaded with 5 teachers, 5 classrooms, 5 subjects, 8 time slots, 6 days
- **Query Capabilities:** Complex JOINs, GROUP BY, aggregation functions
- **GUI Interface:** Tkinter-based professional interface with login and dashboard
- **Export Functionality:** Data can be exported to text files for reporting

KEY FEATURES:

✓ Referential integrity through foreign key constraints

✓ Automatic timestamp tracking (created_at, updated_at)

✓ Unique constraints to prevent duplicate data

✓ Support for complex queries with multiple JOINs

✓ Scalable design for growing data

✓ Easy to query by day, teacher, classroom, or subject

✓ Real-time schedule generation and management

Generated on: 15-11-2025 at 18:35:52