

DATABASE OVERVIEW

Database Name

```
CREATE DATABASE resolveX_grievance_system;  
USE resolveX_grievance_system;
```

TOTAL TABLE COUNT

Category	Tables
Core entities	5
Complaint processing	5
AI & NLP logs	3
Tracking & notifications	3
Analytics & heatmap	3
Security & audit	2
TOTAL	21 TABLES

This is **professional-grade**, not over-engineered, and **academically strong**.

USERS & ACCESS CONTROL

1. users

```
CREATE TABLE users (  
    user_id INT AUTO_INCREMENT PRIMARY KEY,  
    full_name VARCHAR(100),  
    email VARCHAR(100) UNIQUE NOT NULL,  
    password_hash VARCHAR(255) NOT NULL,  
    role ENUM('citizen','department','admin') NOT NULL,  
    is_active BOOLEAN DEFAULT TRUE,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
)
```

2. user_profiles

```
CREATE TABLE user_profiles (
    profile_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT,
    phone VARCHAR(15),
    preferred_language ENUM('English','Tamil','Hindi'),
    FOREIGN KEY (user_id) REFERENCES users(user_id)
);
```

2 DEPARTMENTS & LOCATIONS

3. departments

```
CREATE TABLE departments (
    department_id INT AUTO_INCREMENT PRIMARY KEY,
    department_name VARCHAR(100) UNIQUE,
    description TEXT
);
```

4. locations

```
CREATE TABLE locations (
    location_id INT AUTO_INCREMENT PRIMARY KEY,
    state VARCHAR(100),
    district VARCHAR(100),
    city VARCHAR(100),
    latitude DOUBLE,
    longitude DOUBLE
);
```

 **latitude & longitude are mandatory for Google Heatmap**

3 COMPLAINT CORE TABLES

5. complaints ★ (MOST IMPORTANT)

```
CREATE TABLE complaints (
    complaint_id INT AUTO_INCREMENT PRIMARY KEY,
    complaint_code VARCHAR(30) UNIQUE,
    user_id INT,
    department_id INT,
    location_id INT,
    complaint_text TEXT,
    language ENUM('English','Tamil','Hindi'),
    current_status ENUM(
        'Submitted','Received','Assigned',
        'In Progress','Resolved','Closed'
    ),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(user_id),
    FOREIGN KEY (department_id) REFERENCES departments(department_id),
    FOREIGN KEY (location_id) REFERENCES locations(location_id)
);
```

6. complaint_attachments

```
CREATE TABLE complaint_attachments (
    attachment_id INT AUTO_INCREMENT PRIMARY KEY,
    complaint_id INT,
    file_url TEXT,
    uploaded_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (complaint_id) REFERENCES complaints(complaint_id)
```

);

4 REAL-TIME STATUS TRACKING

7. complaint_status_logs

```
CREATE TABLE complaint_status_logs (
    status_log_id INT AUTO_INCREMENT PRIMARY KEY,
    complaint_id INT,
    old_status VARCHAR(30),
    new_status VARCHAR(30),
    updated_by INT,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (complaint_id) REFERENCES complaints(complaint_id),
    FOREIGN KEY (updated_by) REFERENCES users(user_id)
);
```

📌 Used for:

- Timeline UI
 - Real-time updates
 - Audit trail
-

5 AI / NLP PROCESSING (BERT + Bi-LSTM)

8. nlp_processing_logs

```
CREATE TABLE nlp_processing_logs (
    nlp_id INT AUTO_INCREMENT PRIMARY KEY,
    complaint_id INT,
    bert_model_used VARCHAR(50),
    processed_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (complaint_id) REFERENCES complaints(complaint_id)
);
```

9. sentiment_analysis

```
CREATE TABLE sentiment_analysis (
    sentiment_id INT AUTO_INCREMENT PRIMARY KEY,
    complaint_id INT,
    sentiment_label ENUM('Positive','Negative','Neutral'),
    sentiment_score FLOAT,
    analyzed_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (complaint_id) REFERENCES complaints(complaint_id)
);
```

10 priority_analysis

```
CREATE TABLE priority_analysis (
    priority_id INT AUTO_INCREMENT PRIMARY KEY,
    complaint_id INT,
    urgency_score FLOAT,
    priority_level ENUM('Normal','Urgent'),
    FOREIGN KEY (complaint_id) REFERENCES complaints(complaint_id)
);
```

📌 Derived from:

- LSTM sentiment
 - Keywords
 - Context
-

6 NOTIFICATIONS SYSTEM

11. notifications

```
CREATE TABLE notifications (
    notification_id INT AUTO_INCREMENT PRIMARY KEY,
```

```
user_id INT,  
complaint_id INT,  
message TEXT,  
is_read BOOLEAN DEFAULT FALSE,  
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
FOREIGN KEY (user_id) REFERENCES users(user_id),  
FOREIGN KEY (complaint_id) REFERENCES complaints(complaint_id)  
);
```

7 FEEDBACK & CLOSURE

12. feedback

```
CREATE TABLE feedback (  
feedback_id INT AUTO_INCREMENT PRIMARY KEY,  
complaint_id INT,  
rating INT CHECK (rating BETWEEN 1 AND 5),  
comments TEXT,  
submitted_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
FOREIGN KEY (complaint_id) REFERENCES complaints(complaint_id)  
);
```

8 GOOGLE MAPS HEATMAP SUPPORT

13. heatmap_cache

```
CREATE TABLE heatmap_cache (  
heatmap_id INT AUTO_INCREMENT PRIMARY KEY,  
latitude DOUBLE,  
longitude DOUBLE,  
complaint_count INT,  
last_updated TIMESTAMP DEFAULT CURRENT_TIMESTAMP
```

);

📌 Used to:

- Speed up admin heatmap
 - Avoid recalculating each time
-

14. heatmap_filters

```
CREATE TABLE heatmap_filters (
    filter_id INT AUTO_INCREMENT PRIMARY KEY,
    department_id INT,
    date_from DATE,
    date_to DATE,
    urgency_only BOOLEAN,
    FOREIGN KEY (department_id) REFERENCES departments(department_id)
);
```

💡 ANALYTICS & DASHBOARDS

15. analytics_department_summary

```
CREATE TABLE analytics_department_summary (
    summary_id INT AUTO_INCREMENT PRIMARY KEY,
    department_id INT,
    total_complaints INT,
    resolved_complaints INT,
    avg_resolution_hours FLOAT,
    last_updated TIMESTAMP,
    FOREIGN KEY (department_id) REFERENCES departments(department_id)
);
```

16. analytics_location_summary

```
CREATE TABLE analytics_location_summary (
    location_id INT,
    total_complaints INT,
    urgent_cases INT,
    FOREIGN KEY (location_id) REFERENCES locations(location_id)
);
```

10 SECURITY & AUDIT

17. audit_logs

```
CREATE TABLE audit_logs (
    audit_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT,
    action VARCHAR(255),
    action_time TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(user_id)
);
```

18. login_history

```
CREATE TABLE login_history (
    login_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT,
    login_time TIMESTAMP,
    ip_address VARCHAR(45),
    FOREIGN KEY (user_id) REFERENCES users(user_id)
);
```

1 1 DATASET MANAGEMENT (TRAINING SUPPORT)

19. dataset_registry

```
CREATE TABLE dataset_registry (
    dataset_id INT AUTO_INCREMENT PRIMARY KEY,
    department_id INT,
    language ENUM('English','Tamil','Hindi'),
    record_count INT,
    created_at TIMESTAMP,
    FOREIGN KEY (department_id) REFERENCES departments(department_id)
);
```

20. model_registry

```
CREATE TABLE model_registry (
    model_id INT AUTO_INCREMENT PRIMARY KEY,
    model_name VARCHAR(100),
    model_type ENUM('BERT','Bi-LSTM'),
    language VARCHAR(20),
    version VARCHAR(20),
    trained_on TIMESTAMP
);
```

21. model_predictions

```
CREATE TABLE model_predictions (
    prediction_id INT AUTO_INCREMENT PRIMARY KEY,
    complaint_id INT,
    model_id INT,
    prediction_output VARCHAR(100),
    confidence FLOAT,
    predicted_at TIMESTAMP,
    FOREIGN KEY (complaint_id) REFERENCES complaints(complaint_id),
);
```

```
FOREIGN KEY (model_id) REFERENCES model_registry(model_id)
```

```
);
```