

# **Crime Record & Investigation Database System**

## **Project Report**

---

### **Executive Summary**

This project implements a comprehensive Crime Record and Investigation Database System using PostgreSQL. The system is designed to manage criminal cases, suspects, investigating officers, and evidence with full chain of custody tracking. It provides law enforcement agencies with a robust platform for case management, investigation tracking, and performance analysis.

---

### **1. Project Objectives**

The primary objectives of this database system are:

1. Centralized storage and management of criminal investigation records
  2. Efficient tracking of cases from reporting through resolution
  3. Comprehensive suspect information management with arrest status tracking
  4. Evidence management with automated chain of custody logging
  5. Officer workload distribution and performance monitoring
  6. Generation of analytical reports for decision-making support
  7. Fast query performance through strategic indexing
  8. Data integrity through constraints and automated triggers
-

## 2. Database Architecture

### 2.1 Schema Design

The database follows a normalized relational design with six core tables:

#### Primary Tables:

- **Officers:** Stores law enforcement personnel information including badge numbers, ranks, departments, and contact details
- **Cases:** Central table containing all criminal case information, incident details, status, and severity classifications
- **Suspects:** Individuals connected to cases with biographical information, arrest status, and criminal history
- **Evidence:** Physical and digital evidence items with collection details and storage locations

#### Relationship Tables:

- **Case\_Officers:** Many-to-many relationship tracking officer assignments to cases with roles and active status
- **Evidence\_Chain:** Audit log for evidence custody tracking with timestamps, handlers, and actions

### 2.2 Key Design Decisions

**Normalization:** The schema adheres to Third Normal Form (3NF) to eliminate data redundancy and ensure data integrity.

**Referential Integrity:** Foreign key constraints with CASCADE delete operations ensure orphaned records are automatically removed when parent records are deleted.

**Status Management:** CHECK constraints on status fields ensure only valid values can be entered, preventing data inconsistencies.

**Temporal Tracking:** Timestamp fields with DEFAULT CURRENT\_TIMESTAMP capture action dates automatically.

### **3. Database Features**

#### **3.1 Indexing Strategy**

Strategic indexes have been implemented on high-frequency query fields:

- Case identifiers (case\_id, case\_number)
- Case status and crime type for filtering operations
- Suspect names (first\_name, last\_name) for search functionality
- Officer and case assignments for join operations
- Evidence numbers and case relationships

These indexes significantly improve query performance, particularly for searches involving suspect names and case lookups.

#### **3.2 Database Views**

Four comprehensive views have been created for common reporting requirements:

**Officer Workload View:** Aggregates case assignments per officer, calculating total cases, active cases, solved cases, and solve rate percentages. This view supports management decisions regarding workload distribution.

**Case Summary Report View:** Provides a complete overview of each case including lead officer, suspect count, evidence count, assigned officers, and days the case has been open.

**Unsolved Cases Analysis View:** Focuses on open and cold cases, showing days unsolved, available evidence, and assigned resources to prioritize investigation efforts.

**Evidence Custody View:** Displays the complete chain of custody for evidence items, essential for maintaining legal admissibility in court proceedings.

### **3.3 Automated Triggers**

Two trigger functions automate critical database operations:

**Evidence Chain Logging Trigger:** Automatically creates audit log entries in the evidence\_chain table whenever evidence status is updated. This ensures complete custody tracking without manual intervention.

**Case Status Update Trigger:** Monitors suspect arrest status and automatically updates the parent case to "Solved" when all suspects are convicted, ensuring case status remains current.

---

## **4. Query Capabilities**

The system provides extensive querying capabilities organized into seven categories:

### **4.1 Case Analysis Queries**

- Solved versus unsolved case statistics with percentages
- Case distribution by crime type with solve rates
- High-priority cases requiring immediate attention
- Cold case reports with years unsolved
- Monthly case trends for pattern analysis

### **4.2 Suspect Management Queries**

- Suspect search by name with fuzzy matching
- At-large suspect tracking
- Repeat offender identification across multiple cases
- Arrest status summary statistics

### **4.3 Officer Performance Queries**

- Complete workload reports using the officer\_workload view
- Solve rate rankings to identify top performers
- Detailed case assignment tracking
- Workload imbalance detection for resource reallocation

### **4.4 Evidence Management Queries**

- Evidence summary by case showing collection and analysis status
- Chain of custody reports for specific evidence items
- Evidence with custody issues flagged for review
- Pending analysis tracking to prevent backlogs

## **4.5 Investigation Summary Exports**

- Comprehensive case summary reports
- Active investigation status reports
- Solved cases reports with time-to-resolution metrics
- Department performance dashboards

## **4.6 Specialized Investigation Queries**

- Cases with multiple suspects for complex investigations
- Cases missing critical information (suspects, evidence, or officers)
- Cross-referencing queries for pattern detection

## **4.7 Data Export Queries**

- CSV export functionality for external reporting
  - Integration with business intelligence tools
  - Archival and backup operations
-

## **5. Data Integrity and Security**

### **5.1 Constraints**

The database implements multiple constraint types:

- PRIMARY KEY constraints ensure unique record identification
- FOREIGN KEY constraints maintain referential integrity
- UNIQUE constraints prevent duplicate badge numbers and evidence numbers
- CHECK constraints validate status values and severity levels
- NOT NULL constraints on critical fields prevent incomplete records

### **5.2 Data Validation**

Field-level validation includes:

- Email format validation for officer contacts
- Date consistency (incident dates cannot be in the future)
- Status transitions follow logical progressions
- Numeric ranges for quantitative fields

### **5.3 Audit Trail**

The evidence\_chain table provides a complete audit trail for evidence handling, critical for legal proceedings and internal investigations.

---

## **6. Performance Considerations**

### **6.1 Index Optimization**

Indexes are strategically placed on:

- Frequently joined columns (foreign keys)
- WHERE clause filter columns (status, arrest\_status)
- Search columns (names, case numbers)
- ORDER BY columns (dates)

### **6.2 Query Optimization**

- Views pre-compute complex aggregations
- Subqueries are minimized in favor of joins
- CASE expressions are used for conditional aggregations
- String aggregation uses efficient STRING\_AGG function

### **6.3 Scalability**

The design supports horizontal scaling through:

- Partitioning potential on case dates
  - Archive table strategies for historical data
  - Read replica support for reporting workloads
-

## **7. Sample Data**

The schema includes sample data demonstrating:

- Four officers across different departments and ranks
- Three cases covering robbery, theft, and cybercrime
- Multiple suspects with varying arrest statuses
- Evidence items with initial chain of custody entries
- Officer assignments showing team-based investigations

This sample data enables immediate testing and validation of all queries and features.

---

## **8. Implementation Guidelines**

### **8.1 Deployment Steps**

1. Create PostgreSQL database instance
2. Execute `crime_database_schema.sql` to create all objects
3. Verify table creation and relationships
4. Test triggers with sample data modifications
5. Execute `crime_database_queries.sql` for reporting
6. Configure backup and maintenance schedules

### **8.2 User Roles**

Recommended PostgreSQL roles:

- **dba\_admin**: Full access for database administrators
- **investigator**: Read/write access to cases, suspects, and evidence
- **analyst**: Read-only access for reporting and analysis
- **evidence\_clerk**: Limited access to evidence table only

### **8.3 Maintenance Operations**

Regular maintenance should include:

- Weekly VACUUM ANALYZE for statistics updates
  - Monthly index rebuild for fragmented indexes
  - Quarterly archive of closed cases older than specified retention period
  - Daily automated backups with point-in-time recovery capability
-

## **9. Reporting Capabilities**

The system generates multiple report types:

### **9.1 Management Reports**

- Department performance dashboards with solve rates
- Officer workload distribution for resource allocation
- Case status summaries for executive briefings
- Trend analysis for strategic planning

### **9.2 Operational Reports**

- Daily active case lists for shift briefings
- At-large suspect bulletins
- Evidence pending analysis reports
- Case assignment rosters

### **9.3 Analytical Reports**

- Crime pattern analysis by type and location
  - Solve rate trends over time
  - Officer performance comparisons
  - Case resolution time statistics
-

## **10. Future Enhancements**

Potential system improvements include:

### **10.1 Functional Enhancements**

- Geographic information system (GIS) integration for crime mapping
- Document management system integration for reports and photos
- Mobile application for field officers
- Automated alert system for case milestones
- Integration with external databases (fingerprint, DNA)

### **10.2 Technical Enhancements**

- Full-text search capabilities using PostgreSQL's tsvector
- Time-series analysis for predictive policing
- Machine learning integration for pattern detection
- Data warehouse integration for long-term analytics
- API development for third-party integrations

### **10.3 Security Enhancements**

- Row-level security for multi-jurisdictional access
  - Encryption at rest for sensitive data
  - Audit logging for all data modifications
  - Two-factor authentication integration
  - Role-based access control refinement
-

## **11. Technical Specifications**

**Database Management System:** PostgreSQL 12

**Core Tables:** 6 (Officers, Cases, Suspects, Evidence, CaseOfficers, EvidenceChain)

**Indexes:** 15 strategic indexes for query optimization

**Views:** 4 materialized and standard views

**Triggers:** 2 automated business logic triggers

**Constraints:** 23 constraints (PRIMARY KEY, FOREIGN KEY, CHECK, UNIQUE, NOT NULL)

**Sample Queries:** 30+ queries across 7 functional categories

**Data Types Used:** SERIAL, INTEGER, VARCHAR, TEXT, DATE, TIMESTAMP, BOOLEAN

---

## **12. Conclusion**

The Crime Record and Investigation Database System provides law enforcement agencies with a comprehensive, efficient, and scalable solution for managing criminal investigations. The database design prioritizes data integrity, query performance, and ease of use while maintaining flexibility for future enhancements.

The implemented indexing strategy, automated triggers, and pre-built queries enable investigators and administrators to focus on case work rather than data management. The system's reporting capabilities support both operational decision-making and strategic planning.

With proper implementation, maintenance, and user training, this database system will significantly improve investigation efficiency, case tracking accuracy, and overall departmental performance metrics.

---

## **13. Appendices**

### **Appendix A: File Deliverables**

1. **crime\_database\_schema.sql**: Complete database structure including tables, indexes, views, triggers, and sample data
2. **crime\_database\_queries.sql**: Comprehensive query library for all reporting and analysis needs

### **Appendix B: Installation Commands**

```
psql -U username -d database_name -f crime_database_schema.sql
```

```
psql -U username -d database_name -f crime_database_queries.sql
```

### **Appendix C: Key Performance Indicators**

The system tracks these critical metrics:

- Overall solve rate percentage
  - Average days to case resolution
  - Officer caseload distribution
  - Evidence custody compliance rate
  - Suspect apprehension rate
  - Cold case inventory size
-