# City Database Management

## 1 Introduction to the Mini-World

This database accounts for all the essential and infrastructural bodies required for functioning of a hypothetical city. Modern cities face increasing challenges in managing their resources and services effectively. To address these challenges, a City Management Database has been designed to track and optimize the financial budget of cities, with a primary focus on improving healthcare, and key institutions such as the police and banks. This database serves as a comprehensive solution to ensure data-driven decision-making and better overall urban governance.

## 2 Purpose

The primary purpose of the City Management Database is to centralize, manage, and analyze financial data for cities, enabling government officials, policymakers, and other stakeholders to make informed decisions about resource allocation. By optimizing budget allocation, this database aims to enhance the quality of healthcare, infrastructure, and the efficiency of key institutions.

## 3 Users of the Database

1. Government Officials: City mayors, council members, and department heads can use the database to allocate resources and make informed decisions.

2. Financial Analysts: Professionals analyzing financial trends and making recommendations for budget allocation.

3. Law Enforcement: Federal agencies and law enforcement departments for better investigating crimes and other disputes.

4. Citizens: In some cases, citizens may have limited access to view specific public information, promoting transparency and civic engagement.

# 4 Views of the Database

1. Financial Overview: Provides a summary or high-level view of the city's overall financial health, including revenues, expenditures, and budget allocations for various sectors.
2. Healthcare Management: Offers insights into healthcare budgets, patient data, and medical facility utilization.
3. Federal Institutions: Gives access to personal records of individuals for decisions like offering loans, criminal investigation, and tax-income reports.
4. Institution Efficiency: Evaluates the performance and budget utilization of key institutions like the police and banks.

# 5 Applications of the Database

1. Budget Allocation: The database helps allocate resources efficiently by identifying areas that require increased funding.
2. Policy Development: Policymakers can use data from the database to develop and fine-tune urban policies.
3. Performance Assessment: The database allows tracking and assessment of institutional performance to ensure accountability.
4. Data-Driven Decision-Making: Enables data-driven decision-making by providing timely and accurate information.
5. Public Transparency: Provides citizens with insights into how their tax money is being used and encourages civic engagement.

## 5.1 Secure Access

Access to the database is secured through authentication and authorization mechanisms. Data encryption and strict user access controls ensure that only authorized personnel can view, modify, or add data, safeguarding sensitive financial information.

## 5.2 Strategic Insights

The database provides strategic insights by offering historical data, forecasts, and predictive analytics. Users can identify trends and patterns to make informed decisions for the future, enabling long-term urban planning.

## 5.3 Alleviating Data Redundancy

To reduce data redundancy, the database uses a normalized database design, avoiding data duplication by organizing data into separate tables. For instance, rather than storing the same information about a healthcare facility in multiple places, data on the facility is stored in one central location and referenced where needed. This minimizes the risk of errors and inconsistencies in data.

# 6 Database Requirements

## 6.1 Assumptions

In a practical city landscape, there are myriads of institutions that work together, but for the sake of the project, we keep it simple and focus on the essentials.

## 6.2 Strong Entity Types

### 6.2.1 Citizen

- CIN (Citizen Identification Number): Single attribute, Primary Key, int. This is a unique number given to each citizen to identify them (like SSN/Aadhar/etc).
- Name: Composite Attribute. The name of a person, composed of firstname and lastname.
- DOB (Date Of Birth): Composite Attribute, string. Date of birth is composed of year, month, and day.
- Address: Composite Attribute, string. Door number, Street, Block, Constituency.
- Income: Single attribute, float. The total Income of a person.
- Tax: Single and Derived attribute, float. From the total income of a person, taxes on him are calculated.
- Income Category: Single and Derived attribute, boolean. From the total income of a person, it's decided if the person is above poverty line (APL) or below poverty line (BPL).

### 6.2.2 Neighbourhoods

- PIN (Postal index number): Key attribute, Single attribute, int. Each neighbourhood has a unique postal index number for identification.
- Representative: Single attribute, string. Each neighbourhood has an elected representative in the governing council of the city. They are identified by their CIN.

- Landmark: Multivalued attribute, string. The neighbourhood can have some popular landmarks. Ex: Malls, Amusement parks.
- Population: Single and Derived attribute, Int. The population of the neighbourhood.
- No. of Diseases reported: Single and Derived attribute, Int. The No. of diseases reported per month in the neighbourhood.
- Average Income: Single attribute, float. The average income of the people living in the area.
- Neighbourhood Type: Single attribute, string. Each neighbourhood belongs to a specific type. Example- Residential, Commercial, etc.

# 7 Functional Requirements

## 7.1 Modifications

- **Insert**:
  - When a person is born, insert a new entity into Citizen.
  - A new entity is inserted into Neighborhood if the city gets expanded.
  - When a new hospital is built, insert a new entity into Hospital.
  - For every patient visiting a hospital, the hospital inserts a new entity into Patient.
  - For every customer of a bank, the bank inserts a new entity into Bank Account.
- **Update**:
  - Update the income of a person on a regular basis in People.
  - Update the interest rates as and when they are revised in Bank.
  - Update the population based on the census in People.
- **Delete**:
  - When a person dies, delete the entity in Citizen.
  - When a hospital closes down, delete the entity in Hospital.
  - For every customer closing the bank account, the bank deletes the entity in Bank Account.
  - Upon shutdown of a Law Enforcement Agency, the entity is deleted from LEA.
  - For every crime with justice settled, the LEA deletes the entity from Crime.

# 8 Summary

The City Management Database shines as a beacon of transparency, providing open access to vital financial information for citizens and policymakers. Its robust maintenance ensures data accuracy and security, safeguarding the city's financial health for the long term. By offering insights and historical data, it empowers decision-makers to shape sustainable urban policies. The database's transparency builds trust and accountability, promoting civic engagement and informed decision-making. In this synergy of transparency and maintenance, the database becomes an invaluable asset for the city's long-term well-being.

# 9 Aim of Project

The project consists of two main parts:

## 9.1 Creating the Database

You are required to create your database according to the relational model constructed in phase 3. Ensure to account for all the constraints such as entity, referential, domain, etc. while creating the database. Additionally, load your tables with legitimate values. Each table should contain at least 5 tuples.

## 9.2 Performing Queries on the Database

You need to implement all the functional requirements mentioned in your phase 1 submission using SQL queries. The queries should cover:

### 9.2.1 Modifications

- **Insertion:** At least 2 queries for inserting new records into relevant tables.
- **Update:** At least 2 queries for updating existing records in relevant tables.
- **Deletion:** At least 2 queries for deleting records from relevant tables.

### 9.2.2 Retrievals

- **Selection:** At least 2 queries for selecting specific records based on certain conditions.
- **Projection:** At least 2 queries for projecting specific attributes from relevant tables.
- **Aggregation:** At least 2 queries for aggregating data (e.g., sum, count, average).

- **Search:** At least 2 queries for searching and retrieving specific information based on criteria.
- **Analysis:** At least 2 queries involving the JOIN clause to analyze relationships between entities and tables.