

# A Project Report On Smart City Management System

DEVELOPED BY:  
IT146 – VIVEK THUMAR  
IT157 – VENU VIRPARIA

Guided By  
Internal Guide:  
Prof. Shweta Jambukia

Department of Information Technology  
Faculty of Technology  
DDU



Department of Information Technology Faculty of Technology,  
Dharmsinh Desai University College Road, Nadiad-  
387001

## CERTIFICATE

This is to certify that the project entitled “Smart City Management System” is a bonafide report of the work carried out by

1) Vivek Thumar Student ID No: 22ITUOS111

2) Venu Virparia Student ID No: 22ITUOS055

of Department of Information Technology, semester IV, under the guidance and supervision for the subject Database Management System. They were involved in Project training during the academic year 2023-2024.

Prof. Shweta Jambukia Project Guide, Department of Information Technology, Faculty of Technology, Dharmsinh Desai University, Nadiad  
Date: 04/03/2024

Prof. Vipul Dabhi Head, Department of Information Technology

# INDEX

I. Title Page.....	I
II. Certificate.....	II
1.SYSTEM OVERVIEW.....	4
a. Current system.....	4
b. Advantages of the Proposed system (over current) .....	4
2.E-R Diagram.....	5
3.RELATIONAL SCHEMA .....	6
4.DATA DICTIONARY.....	7
5.DATABASE IMPLEMENTATION.....	11
5.1 Create Schema .....	11
5.2 Insert Data values.....	14
5.3 Insertion Output.....	21
6. QUERIES.....	26
6.1 Queries on basic concepts .....	26
6.2 Queries on join, subquery, group by.....	27
6.2 PL/SQL Blocks (Views).....	30
6.4 Functions & Triggers.....	31
6.5 Cursors.....	34

## 1.SYSTEM OVERVIEW

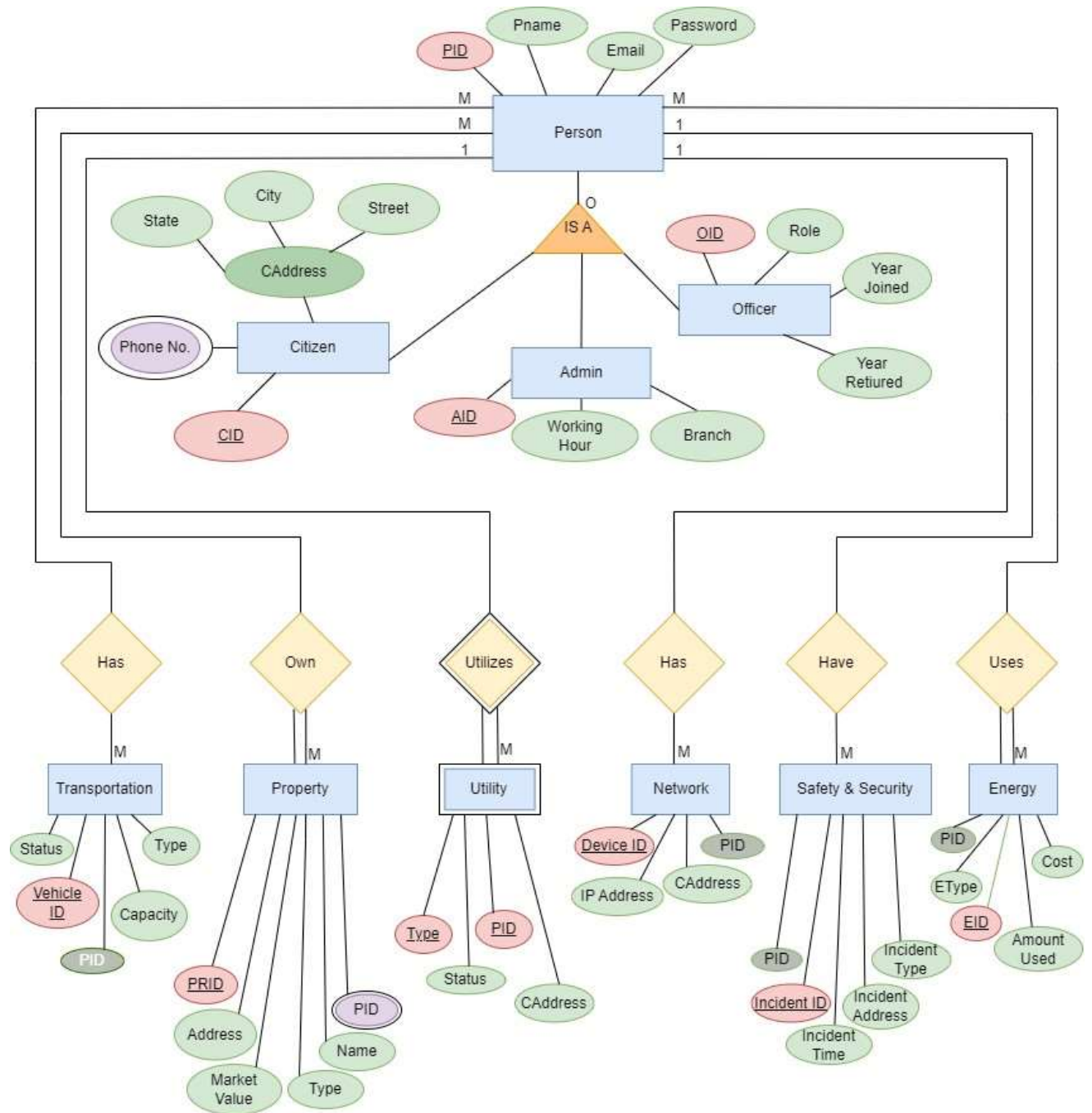
The Smart City Management System (SCMS) is a comprehensive web-based platform designed to efficiently manage various aspects of city operations. It serves as a centralized hub for collecting and analysing data related to urban activities, enabling better decision-making and resource allocation.

With SCMS, users can input and track information pertaining to key city functions such as transportation, utilities, public safety, environmental management, and citizen services. The system facilitates the monitoring of city infrastructure, including roads, public spaces, and utilities, ensuring timely maintenance and repairs.

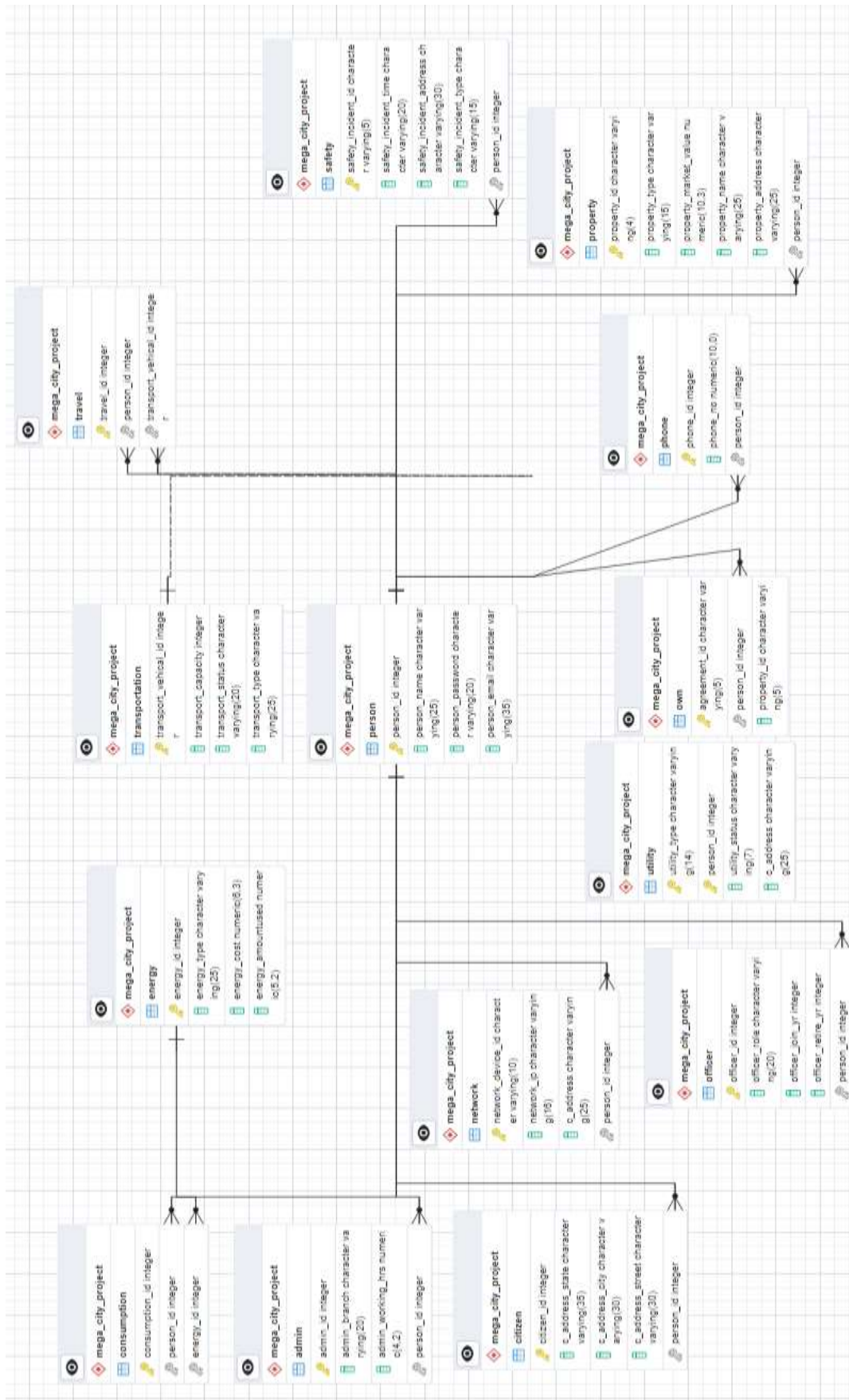
### 1.3 ADVANTAGES OF THE PROPOSED SYSTEM

1. **Improved Efficiency:** SCMS streamlines city operations by centralizing data and processes, reducing administrative time and enhancing overall efficiency.
2. **Enhanced Resource Allocation:** By providing insights into city-wide trends and demands, SCMS enables more informed decision-making regarding resource allocation, leading to optimized service delivery.
3. **Effective Infrastructure Management:** With SCMS, city administrators can monitor the condition of infrastructure assets in real-time, enabling proactive maintenance and minimizing disruptions to residents.
4. **Data-Driven Decision Making:** By analysing data collected from various sources, SCMS enables evidence-based decision-making, allowing city administrators to address challenges and capitalize on opportunities more effectively.

## 2. ENTITY-RELATIONSHIP MODEL



### 3. RELATIONAL SCHEMA



## 4. Data Dictionary

### 4.1 Person

```
DBMS_FINAL_PROJECT=# \d mega_city_project.person
Table "mega_city_project.person"
  Column          | Type          | Collation | Nullable | Default
-----|-----|-----|-----|-----
 person_id        | integer       |           | not null |
 person_name      | character varying(25) |           | not null |
 person_password   | character varying(20) |           | not null |
 person_email     | character varying(35) |           | not null |
Indexes:
    "person_pkey" PRIMARY KEY, btree (person_id)
Referenced by:
    TABLE "mega_city_project.admin" CONSTRAINT "admin_person_id_fkey" FOREIGN KEY (person_id) REFERENCES mega_city_project.person(person_id)
    TABLE "mega_city_project.citizen" CONSTRAINT "citizen_person_id_fkey" FOREIGN KEY (person_id) REFERENCES mega_city_project.person(person_id)
    TABLE "mega_city_project.consumption" CONSTRAINT "consumption_person_id_fkey" FOREIGN KEY (person_id) REFERENCES mega_city_project.person(person_id)
    TABLE "mega_city_project.network" CONSTRAINT "network_person_id_fkey" FOREIGN KEY (person_id) REFERENCES mega_city_project.person(person_id)
    TABLE "mega_city_project.officer" CONSTRAINT "officer_person_id_fkey" FOREIGN KEY (person_id) REFERENCES mega_city_project.person(person_id)
    TABLE "mega_city_project.own" CONSTRAINT "own_person_id_fkey" FOREIGN KEY (person_id) REFERENCES mega_city_project.person(person_id)
    TABLE "mega_city_project.phone" CONSTRAINT "phone_person_id_fkey" FOREIGN KEY (person_id) REFERENCES mega_city_project.person(person_id)
    TABLE "mega_city_project.safety" CONSTRAINT "safety_person_id_fkey" FOREIGN KEY (person_id) REFERENCES mega_city_project.person(person_id)
    TABLE "mega_city_project.travel" CONSTRAINT "travel_person_id_fkey" FOREIGN KEY (person_id) REFERENCES mega_city_project.person(person_id)
DBMS_FINAL_PROJECT=# |
```

### 4.2 Citizen

```
DBMS_FINAL_PROJECT=# \d mega_city_project.citizen
Table "mega_city_project.citizen"
  Column          | Type          | Collation | Nullable | Default
-----|-----|-----|-----|-----
 citizen_id       | integer       |           | not null |
 c_address_state  | character varying(35) |           | not null |
 c_address_city   | character varying(30) |           | not null |
 c_address_street | character varying(30) |           | not null |
 person_id        | integer       |           | not null |
Indexes:
    "citizen_pkey" PRIMARY KEY, btree (citizen_id)
Foreign-key constraints:
    "citizen_person_id_fkey" FOREIGN KEY (person_id) REFERENCES mega_city_project.person(person_id)
DBMS_FINAL_PROJECT=# |
```

### 4.3 Admin

```
DBMS_FINAL_PROJECT=# \d mega_city_project.admin
Table "mega_city_project.admin"
  Column          | Type          | Collation | Nullable | Default
-----|-----|-----|-----|-----
 admin_id         | integer       |           | not null |
 admin_branch     | character varying(20) |           | not null |
 admin_working_hrs | numeric(4,2)   |           | not null |
 person_id        | integer       |           | not null |
Indexes:
    "admin_pkey" PRIMARY KEY, btree (admin_id)
Foreign-key constraints:
    "admin_person_id_fkey" FOREIGN KEY (person_id) REFERENCES mega_city_project.person(person_id)
DBMS_FINAL_PROJECT=# |
```

## 4.4 Officer

```
DBMS_FINAL_PROJECT-# \d mega_city_project.officer
Table "mega_city_project.officer"
  Column      | Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
officer_id   | integer       |           | not null |
officer_role | character_varying(20) |           | not null |
officer_join_yr | integer       |           | not null |
officer_retire_yr | integer       |           | not null |
person_id    | integer       |           | not null |

Indexes:
  "officer_pkey" PRIMARY KEY, btree (officer_id)
Foreign-key constraints:
  "officer_person_id_fkey" FOREIGN KEY (person_id) REFERENCES mega_city_project.person(person_id)

DBMS_FINAL_PROJECT-# |
```

## 4.5 Phone

```
DBMS_FINAL_PROJECT-# \d mega_city_project.phone
Table "mega_city_project.phone"
  Column      | Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
phone_id     | integer       |           | not null |
phone_no     | numeric(10,0) |           | not null |
person_id    | integer       |           | not null |

Indexes:
  "phone_pkey" PRIMARY KEY, btree (phone_id)
Foreign-key constraints:
  "phone_person_id_fkey" FOREIGN KEY (person_id) REFERENCES mega_city_project.person(person_id)
Triggers:
  phone_trigger BEFORE INSERT OR UPDATE ON mega_city_project.phone FOR EACH ROW EXECUTE FUNCTION phone_trigger_function()

DBMS_FINAL_PROJECT-# |
```

## 4.6 Transportation

```
DBMS_FINAL_PROJECT-# \d mega_city_project.transportation
Table "mega_city_project.transportation"
  Column          | Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
transport_vehical_id | integer       |           | not null |
transport_capacity   | integer       |           | not null |
transport_status     | character_varying(20) |           | not null |
transport_type       | character_varying(25) |           | not null |

Indexes:
  "transportation_pkey" PRIMARY KEY, btree (transport_vehical_id)
Referenced by:
  TABLE "mega_city_project.travel" CONSTRAINT "travel_transport_vehical_id_fkey" FOREIGN KEY (transport_vehical_id) REFERENCES mega_city_project.transportation(transport_vehical_id)

DBMS_FINAL_PROJECT-# |
```

## 4.7 Travel

```
DBMS_FINAL_PROJECT-# \d mega_city_project.travel
Table "mega_city_project.travel"
  Column          | Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
travel_id        | integer       |           | not null |
person_id        | integer       |           | not null |
transport_vehical_id | integer       |           | not null |

Indexes:
  "travel_pkey" PRIMARY KEY, btree (travel_id)
Foreign-key constraints:
  "travel_person_id_fkey" FOREIGN KEY (person_id) REFERENCES mega_city_project.person(person_id)
  "travel_transport_vehical_id_fkey" FOREIGN KEY (transport_vehical_id) REFERENCES mega_city_project.transportation(transport_vehical_id)

DBMS_FINAL_PROJECT-# |
```



## 4.8 Energy

```
DBMS_FINAL_PROJECT=# \d mega_city_project.energy
Table "mega_city_project.energy"
  Column      | Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
energy_id     | integer       |           | not null |
energy_type   | character varying(25) |           | not null |
energy_cost   | numeric(6,1)  |           | not null |
energy_amountused | numeric(5,2)  |           | not null |
Indexes:
    "energy_pkey" PRIMARY KEY, btree (energy_id)
Referenced by:
    TABLE "mega_city_project.consumption" CONSTRAINT "consumption_energy_id_fkey" FOREIGN KEY (energy_id) REFERENCES mega_city_project.energy(energy_id)

DBMS_FINAL_PROJECT=# |
```

## 4.9 Consumption

```
DBMS_FINAL_PROJECT=# \d mega_city_project.consumption
Table "mega_city_project.consumption"
  Column      | Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
consumption_id | integer       |           | not null |
person_id      | integer       |           | not null |
energy_id      | integer       |           | not null |
Indexes:
    "consumption_pkey" PRIMARY KEY, btree (consumption_id)
Foreign-key constraints:
    "consumption_energy_id_fkey" FOREIGN KEY (energy_id) REFERENCES mega_city_project.energy(energy_id)
    "consumption_person_id_fkey" FOREIGN KEY (person_id) REFERENCES mega_city_project.person(person_id)

DBMS_FINAL_PROJECT=# |
```

## 4.10 Network

```
DBMS_FINAL_PROJECT=# \d mega_city_project.network
Table "mega_city_project.network"
  Column      | Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
network_device_id | character varying(10) |           | not null |
network_ip        | character varying(16) |           | not null |
c_address         | character varying(25) |           | not null |
person_id         | integer         |           | not null |
Indexes:
    "network_pkey" PRIMARY KEY, btree (network_device_id)
Foreign-key constraints:
    "network_person_id_fkey" FOREIGN KEY (person_id) REFERENCES mega_city_project.person(person_id)

DBMS_FINAL_PROJECT=# |
```

## 4.11 Property

```
DBMS_FINAL_PROJECT=# \d mega_city_project.property
Table "mega_city_project.property"
  Column      | Type          | Collation | Nullable | Default
-----+-----+-----+-----+-----
property_id   | character varying(4) |           | not null |
property_type | character varying(15) |           | not null |
property_market_value | numeric(10,3) |           | not null |
property_name | character varying(25) |           | not null |
property_address | character varying(25) |           | not null |
Indexes:
    "property_pkey" PRIMARY KEY, btree (property_id)
Triggers:
    display_deleted_property AFTER DELETE ON mega_city_project.property FOR EACH ROW EXECUTE FUNCTION display_deleted_property_function()

DBMS_FINAL_PROJECT=# |
```

## 4.12 Own

```
DBMS_FINAL_PROJECT=# \d mega_city_project.own
Table "mega_city_project.own"
  Column      |      Type      | Collation | Nullable | Default
-----+-----+-----+-----+-----
agreement_id | character varying(5) |           | not null |
person_id    | integer          |           |          |
property_id  | character varying(5) |           |          |
Indexes:
    "own_pkey" PRIMARY KEY, btree (agreement_id)
Foreign-key constraints:
    "own_person_id_fkey" FOREIGN KEY (person_id) REFERENCES mega_city_project.person(person_id)

DBMS_FINAL_PROJECT=# |
```

## 4.13 Utility

```
DBMS_FINAL_PROJECT=# \d mega_city_project.utility
Table "mega_city_project.utility"
  Column      |      Type      | Collation | Nullable | Default
-----+-----+-----+-----+-----
utility_type  | character varying(14) |           | not null |
person_id    | integer          |           | not null |
utility_status | character varying(7) |           | not null |
c_address     | character varying(25) |           | not null |
Indexes:
    "utility_pkey" PRIMARY KEY, btree (utility_type, person_id)

DBMS_FINAL_PROJECT=# |
```

## 4.14 Safety

```
DBMS_FINAL_PROJECT=# \d mega_city_project.safety
Table "mega_city_project.safety"
  Column      |      Type      | Collation | Nullable | Default
-----+-----+-----+-----+-----
safety_incident_id | character varying(5) |           | not null |
safety_incident_time | character varying(20) |           | not null |
safety_incident_address | character varying(30) |           | not null |
safety_incident_type | character varying(15) |           | not null |
person_id        | integer          |           | not null |
Indexes:
    "safety_pkey" PRIMARY KEY, btree (safety_incident_id)
Foreign-key constraints:
    "safety_person_id_fkey" FOREIGN KEY (person_id) REFERENCES mega_city_project.person(person_id)
Triggers:
    safety_trigger AFTER INSERT OR UPDATE ON mega_city_project.safety FOR EACH ROW EXECUTE FUNCTION safety_trigger_function()

DBMS_FINAL_PROJECT=# |
```

## 5. DATA IMPLEMENTATION

### A) SCHEMA

#### 5.1.1 Person

```
CREATE TABLE mega_city_project.person (  
  person_id INT PRIMARY KEY NOT NULL,    person_name  
  VARCHAR(25) NOT NULL,    person_password VARCHAR(20)  
  NOT NULL,    person_email VARCHAR(35) NOT NULL  
);
```

#### 5.1.2 Citizen

```
CREATE TABLE  
  mega_city_project.citizen (  
    citizen_id INT PRIMARY KEY NOT NULL,  
  c_address_state VARCHAR(35) NOT NULL,  
  c_address_city VARCHAR(30) NOT NULL,  
  c_address_street VARCHAR(30) NOT NULL,  
    person_id INT NOT NULL,  
    FOREIGN KEY (person_id) REFERENCES mega_city_project.person  
  (person_id)  
);
```

#### 5.1.3 Admin

```
CREATE TABLE  
  mega_city_project.admin (  
    admin_id INT PRIMARY KEY NOT NULL,  
  admin_branch VARCHAR(20) NOT NULL,  
  admin_working_hrs NUMERIC(4, 2) NOT NULL,  
    person_id INT NOT NULL,  
    FOREIGN KEY (person_id) REFERENCES mega_city_project.person  
  (person_id)  
);
```

#### 5.1.4 Officer

```
CREATE TABLE
  mega_city_project.officer (
    officer_id INT PRIMARY KEY NOT NULL,
    officer_role VARCHAR(20) NOT NULL,
    officer_join_yr INT NOT NULL,
    officer_retire_yr INT NOT NULL,
    person_id INT NOT NULL,
    FOREIGN KEY (person_id) REFERENCES mega_city_project.person
    (person_id)
  );
```

#### 5.1.5 Phone

```
CREATE TABLE
  mega_city_project.phone (
    phone_id INT PRIMARY KEY NOT NULL,
    phone_no NUMERIC(10) NOT NULL,
    person_id INT NOT NULL,
    FOREIGN KEY (person_id) REFERENCES mega_city_project.person
    (person_id)
  );
```

#### 5.1.6 Transportation

```
CREATE TABLE
  mega_city_project.transportation (
    transport_vehical_id INT
    PRIMARY KEY NOT NULL,
    transport_capacity INT NOT NULL,
    transport_status VARCHAR(20) NOT NULL,
    transport_type
    VARCHAR(25) NOT NULL
  );
```

#### 5.1.7 Travel

```
CREATE TABLE
  mega_city_project.travel (
    travel_id INT PRIMARY KEY NOT NULL,
    person_id INT NOT NULL,
    transport_vehical_id INT NOT NULL,
    FOREIGN KEY (person_id) REFERENCES mega_city_project.person
    (person_id),
    FOREIGN KEY (transport_vehical_id) REFERENCES
    mega_city_project.transportation (transport_vehical_id)
  );
```

### 5.1.8 Energy

```
CREATE TABLE mega_city_project.energy ( energy_id INT
PRIMARY KEY NOT NULL, energy_type VARCHAR(25) NOT NULL,
energy_cost NUMERIC(6, 3) NOT NULL, energy_amountused
NUMERIC(5, 2) NOT NULL
);
```

### 5.1.9 Consumption

```
CREATE TABLE
mega_city_project.consumption (
consumption_id INT PRIMARY KEY NOT NULL,
person_id INT NOT NULL, energy_id INT
NOT NULL,
FOREIGN KEY (person_id) REFERENCES mega_city_project.person
(person_id),
FOREIGN KEY (energy_id) REFERENCES mega_city_project.energy
(energy_id)
);
```

### 5.1.10 Network

```
CREATE TABLE
mega_city_project.network (
network_device_id VARCHAR(10) PRIMARY KEY NOT NULL,
network_ip VARCHAR(16) NOT NULL,
c_address VARCHAR(25) NOT NULL,
person_id INT NOT NULL,
FOREIGN KEY (person_id) REFERENCES mega_city_project.person
(person_id)
);
```

### 5.1.11 Property

```
CREATE TABLE
mega_city_project.property (
property_id VARCHAR(4) NOT NULL PRIMARY KEY, property_type
VARCHAR(15) NOT NULL, property_market_value NUMERIC(10, 3) NOT
NULL, property_name VARCHAR(25) NOT NULL, property_address
VARCHAR(25) NOT NULL
);
```

### 5.1.12 Own

```
CREATE TABLE
    mega_city_project.own (
        agreement_id VARCHAR(5) PRIMARY KEY NOT NULL,
        person_id INT,
        property_id VARCHAR(5),
        FOREIGN KEY (person_id) REFERENCES mega_city_project.person
        (person_id),
        FOREIGN KEY (property_id) REFERENCES
        mega_city_project.property (property_id)
    );
```

### 5.1.13 Utility

```
CREATE TABLE
    mega_city_project.utility (    utility_type
    VARCHAR(14) NOT NULL,    person_id INT NOT NULL,
        PRIMARY KEY (utility_type, person_id),
    utility_status VARCHAR(7) NOT NULL,    c_address
    VARCHAR(25) NOT NULL
    );
```

### 5.1.14 Safety

```
CREATE TABLE
    mega_city_project.safety (
        safety_incident_id VARCHAR(5) NOT NULL PRIMARY KEY,
        safety_incident_time VARCHAR(20) NOT NULL,
        safety_incident_address VARCHAR(30) NOT NULL,
        safety_incident_type VARCHAR(15) NOT NULL,
        person_id INT NOT NULL,
        FOREIGN KEY (person_id) REFERENCES mega_city_project.person
        (person_id)
    );
```

## B) DATA INSERTION

### 5.2.1 Person

```
INSERT INTO
    mega_city_project.person (person_id, person_name,
    person_password, person_email)
VALUES
    (1, 'Rajesh Kumar', 'pass123', 'rajesh.kumar@email.com'),
    (2, 'Priya Sharma', 'priya456', 'priya.sharma@email.com'),
```

```
(3,'Vikram Singh','vikram789','vikram.singh@email.com'),
(4,'Amit Patel','secure123','amit.patel@email.com'),
(5,'Neha Sharma','pass789','neha.sharma@email.com'),
(6,'Raj Singh','strongpwd456','raj.singh@email.com'),
(7,'Pooja Verma','pooja123','pooja.verma@email.com'),
(8,'Vikram Gupta','vikram789','vikram.gupta@email.com'),
(9,'Anita Kapoor','anita456','anita.kapoor@email.com');
```

### 5.2.2 Citizen

```
INSERT INTO
mega_city_project.citizen (citizen_id, c_address_state,
c_address_city, c_address_street, person_id)
VALUES
(501, 'Maharashtra', 'Mumbai', 'Navi Mumbai', 1),
(502, 'Uttar Pradesh', 'Lucknow', 'Gomti Nagar', 2),
(503, 'Karnataka', 'Bangalore', 'MG Road', 3),
(504, 'Karnataka', 'Bangalore', 'Brigade Road', 4),
(505, 'Maharashtra', 'Mumbai', 'Juhu Beach', 5),
(506, 'Uttar Pradesh', 'Lucknow', 'Hazratganj', 6),
(507, 'Rajasthan', 'Jaipur', 'MI Road', 7),
(508, 'Gujarat', 'Ahmedabad', 'Ashram Road', 8),
(509, 'Tamil Nadu', 'Chennai', 'Mount Road', 9);
```

### 5.2.3 Admin

```
INSERT INTO
mega_city_project.admin (admin_id, admin_branch,
admin_working_hrs, person_id)
VALUES
(101, 'Public Works', 40.0, 1),
(102, 'Finance', 35.5, 2),
(103, 'Health', 38.0, 3),
(104, 'Public Relations', 40.0, 4),
(105, 'Education', 35.5, 5),
(106, 'Transportation', 38.0, 6),
(107, 'Environment', 37.0, 7),
(108, 'Technology', 36.5, 8),
(109, 'Emergency Services', 39.5, 9);
```

### 5.2.4 Officer

```
INSERT INTO
  mega_city_project.officer (officer_id, officer_role,
officer_join_yr, officer_retire_yr, person_id)
VALUES
  (1, 'Security Officer', 2010, 2025, 1),
  (2, 'Urban Planner', 2008, 2030, 2),
  (3, 'IT Specialist', 2015, 2028, 3),
  (4, 'Traffic Management', 2012, 2024, 4),
  (5, 'City Architect', 2011, 2030, 5),
  (6, 'Energy Analyst', 2016, 2028, 6),
  (7, 'Health Inspector', 2013, 2027, 7),
  (8, 'IT Support', 2018, 2029, 8),
  (9, 'Safety Officer', 2014, 2032, 9);
```

### 5.2.5 Phone

```
INSERT INTO
  mega_city_project.phone (phone_id, phone_no, person_id)
VALUES
  (601, 9876543210, 1),
  (602, 8765432109, 2),
  (603, 7890123456, 3),
  (604, 9876543211, 4),
  (605, 8765432108, 5),
  (606, 7890123457, 6),
  (607, 8765432106, 7),
  (608, 9876543215, 8),
  (609, 7890123454, 9);
```

### 5.2.6 Transportation

```
INSERT INTO
  mega_city_project.transportation (transport_vehical_id,
transport_capacity, transport_status, transport_type)
VALUES
  (101, 50, 'Active', 'Bus'),
  (102, 30, 'Inactive', 'Car'),
  (103, 75, 'Active', 'Train'),
  (104, 40, 'Active', 'Metro'),
  (105, 25, 'Inactive', 'Bicycle'),
  (106, 60, 'Active', 'Tram'),
  (107, 35, 'Inactive', 'Rickshaw'),
  (108, 50, 'Active', 'Electric Scooter'),
  (109, 70, 'Active', 'Monorail');
```



### 5.2.7 Travel

```
INSERT INTO
  mega_city_project.travel (travel_id, person_id,
transport_vehical_id)
VALUES
  (201, 1, 101),
  (202, 2, 103),
  (203, 3, 102),
  (204, 4, 104),
  (205, 5, 108),
  (206, 6, 107),
  (207, 7, 106),
  (208, 8, 105),
  (209, 9, 109);
```

### 5.2.8 Energy

```
INSERT INTO
  mega_city_project.energy (energy_id, energy_type, energy_cost,
energy_amountused)
VALUES
  (301, 'Electricity', 0.12, 150.25),
  (302, 'Gas', 1.5, 75.5),
  (303, 'Solar', 0, 200.0),
  (304, 'Solar', 0, 180.75),
  (305, 'Wind', 0, 150.0),
  (306, 'Electricity', 0.15, 120.5),
  (307, 'Hydro', 0, 200.0),
  (308, 'Bioenergy', 0.1, 90.25),
  (309, 'Gas', 1.2, 75.0);
```

### 5.2.9 Consumption

```
INSERT INTO
  mega_city_project.consumption (consumption_id, person_id,
energy_id)
VALUES
  (401, 1, 301),
  (402, 2, 302),
  (403, 3, 303),
  (404, 4, 304),
  (405, 5, 306),
  (406, 6, 308),
  (407, 7, 305),
```

```
(408, 8, 309),  
(409, 9, 307);
```

### 5.2.10 Network

```
INSERT INTO  
  mega_city_project.network (network_device_id, network_ip,  
c_address, person_id)  
VALUES  
  ('NW001', '192.168.1.1', 'Navi Mumbai', 1),  
  ('NW002', '192.168.1.2', 'Gomti Nagar', 2),  
  ('NW003', '192.168.1.3', 'MG Road', 3),  
  ('NW004', '192.168.1.4', 'Brigade Road', 4),  
  ('NW005', '192.168.1.5', 'Juhu Beach', 5),  
  ('NW006', '192.168.1.6', 'Hazratganj', 6),  
  ('NW007', '192.168.1.7', 'MI Road', 7),  
  ('NW008', '192.168.1.8', 'Ashram Road', 8),  
  ('NW009', '192.168.1.9', 'Mount Road', 9);
```

### 5.2.11 Property

```
INSERT INTO  
  mega_city_project.property (property_id, property_type,  
property_market_value, property_name, property_address)  
VALUES  
  ('P001', 'Residential', 2500000.0, 'Green Heights', 'Palm  
Beach Road'),  
  ('P002', 'Commercial', 5000000.0, 'Tech Park Plaza', 'Silk  
Board'),  
  ('P003', 'Industrial', 7500000.0, 'Manufacturing Hub',  
'Industrial Area'),  
  ('P004', 'Residential', 2800000.0, 'Sunrise Apartments',  
'Sunset Boulevard'),  
  ('P005', 'Commercial', 5500000.0, 'Corporate Towers', 'Tech  
Park Avenue'),  
  ('P006', 'Industrial', 8000000.0, 'Manufacturing Complex 2',  
'Industrial Zone 2'),  
  ('P007', 'Residential', 3200000.0, 'Green Gardens', 'Rose  
Street'),  
  ('P008', 'Commercial', 6000000.0, 'Business Hub', 'Corporate  
Avenue'),  
  ('P009', 'Industrial', 9000000.0, 'Tech Manufacturing', 'Tech  
Zone 3');
```

### 5.2.12 Own

```
INSERT INTO
  mega_city_project.own (agreement_id, person_id, property_id)
VALUES
  ('A001', 1, 'P001'),
  ('A002', 2, 'P002'),
  ('A003', 3, 'P003'),
  ('A004', 4, 'P004'),
  ('A005', 5, 'P005'),
  ('A006', 6, 'P006'),
  ('A007', 7, 'P007'),
  ('A008', 8, 'P008'),
  ('A009', 9, 'P009');
```

### 5.2.13 Utility

```
INSERT INTO
  mega_city_project.utility (utility_type, person_id,
utility_status, c_address)
VALUES
  ('Water', 1, 'Active', 'Residential Area'),
  ('Electricity', 2, 'Active', 'Business District'),
  ('Internet', 3, 'Active', 'Tech Zone'),
  ('Water', 4, 'Active', 'Residential Area 3'),
  ('Electricity', 5, 'Active', 'Business District 3'),
  ('Internet', 6, 'Active', 'Tech Zone 3'),
  ('Water', 7, 'Active', 'Residential Area 2'),
  ('Electricity', 8, 'Active', 'Business District 2'),
  ('Internet', 9, 'Active', 'Tech Zone 2');
```

### 5.2.13 Safety

```
INSERT INTO
  mega_city_project.safety (safety_incident_id,
safety_incident_time, safety_incident_address,
safety_incident_type, person_id)
VALUES
  ('S001', '2023-05-10 15:30', 'Palm Beach Road', 'Fire', 2),
  ('S002', '2023-07-22 09:45', 'Silk Board', 'Flood', 3),
  ('S003', '2024-01-05 12:15', 'Industrial Area', 'Earthquake',
1),
  ('S004', '2023-05-10 16:30', 'Sunset Boulevard', 'Fire', 5),
```

```
('S005', '2023-07-22 10:45', 'Tech Park Avenue', 'Flood', 6),  
('S006', '2024-01-05 13:30', 'Industrial Zone 2',  
'Earthquake', 4),  
('S007', '2024-02-15 14:20', 'Rose Street', 'Fire', 7),  
('S008', '2024-03-20 08:10', 'Corporate Avenue', 'Flood', 8),  
('S009', '2024-04-05 16:45', 'Tech Zone 3', 'Earthquake', 9);
```

## C) INSERTION OUTPUT:

### 5.3.1 Person

```
DBMS_FINAL_PROJECT=# SELECT *FROM mega_city_project.person;
```

person_id	person_name	person_password	person_email
1	Rajesh Kumar	pass123	rajesh.kumar@email.com
2	Priya Sharma	priya456	priya.sharma@email.com
3	Vikram Singh	vikram789	vikram.singh@email.com
4	Amit Patel	secure123	amit.patel@email.com
5	Neha Sharma	pass789	neha.sharma@email.com
6	Raj Singh	strongpwd456	raj.singh@email.com
7	Pooja Verma	pooja123	pooja.verma@email.com
8	Vikram Gupta	vikram789	vikram.gupta@email.com
9	Anita Kapoor	anita456	anita.kapoor@email.com

(9 rows)

```
DBMS_FINAL_PROJECT=# |
```

### 5.3.2 Citizen

```
DBMS_FINAL_PROJECT=# SELECT *FROM mega_city_project.citizen;
```

citizen_id	c_address_state	c_address_city	c_address_street	person_id
501	Maharashtra	Mumbai	Navi Mumbai	1
502	Uttar Pradesh	Lucknow	Gomti Nagar	2
503	Karnataka	Bangalore	MG Road	3
504	Karnataka	Bangalore	Brigade Road	4
505	Maharashtra	Mumbai	Juhu Beach	5
506	Uttar Pradesh	Lucknow	Hazratganj	6
507	Rajasthan	Jaipur	MI Road	7
508	Gujarat	Ahmedabad	Ashram Road	8
509	Tamil Nadu	Chennai	Mount Road	9

(9 rows)

```
DBMS_FINAL_PROJECT=# |
```

### 5.3.3 Admin

```
DBMS_FINAL_PROJECT=# SELECT *FROM mega_city_project.admin;
```

admin_id	admin_branch	admin_working_hrs	person_id
101	Public Works	40.00	1
102	Finance	35.50	2
103	Health	38.00	3
104	Public Relations	40.00	4
105	Education	35.50	5
106	Transportation	38.00	6
107	Environment	37.00	7
108	Technology	36.50	8
109	Emergency Services	39.50	9

(9 rows)

```
DBMS_FINAL_PROJECT=# |
```

### 5.3.4 Officer

```
DBMS_FINAL_PROJECT=# SELECT *FROM mega_city_project.officer;
```

officer_id	officer_role	officer_join_yr	officer_retire_yr	person_id
1	Security Officer	2010	2025	1
2	Urban Planner	2008	2030	2
3	IT Specialist	2015	2028	3
4	Traffic Management	2012	2024	4
5	City Architect	2011	2030	5
6	Energy Analyst	2016	2028	6
7	Health Inspector	2013	2027	7
8	IT Support	2018	2029	8
9	Safety Officer	2014	2032	9

(9 rows)

```
DBMS_FINAL_PROJECT=#
```

### 5.3.5 Phone

```
DBMS_FINAL_PROJECT=# SELECT *FROM mega_city_project.phone;
```

phone_id	phone_no	person_id
601	9876543210	1
602	8765432109	2
603	7890123456	3
604	9876543211	4
605	8765432108	5
606	7890123457	6
607	8765432106	7
608	9876543215	8
609	7890123454	9
610	123456789	7
234	123456789	5
200	123456789	1
202	123456789	3
777	123456789	7

(14 rows)

```
DBMS_FINAL_PROJECT=#
```

### 5.3.6 Transportation

```
DBMS_FINAL_PROJECT=# SELECT *FROM mega_city_project.transportation;
```

transport_vehical_id	transport_capacity	transport_status	transport_type
101	50	Active	Bus
102	30	Inactive	Car
103	75	Active	Train
104	40	Active	Metro
105	25	Inactive	Bicycle
106	60	Active	Tram
107	35	Inactive	Rickshaw
108	50	Active	Electric Scooter
109	70	Active	Monorail

(9 rows)

```
DBMS_FINAL_PROJECT=#
```

### 5.3.7 Travel

```
DBMS_FINAL_PROJECT=# SELECT *FROM mega_city_project.travel;
 travel_id | person_id | transport_vehical_id
-----+-----+-----
      201 |         1 |                101
      202 |         2 |                103
      203 |         3 |                102
      204 |         4 |                104
      205 |         5 |                108
      206 |         6 |                107
      207 |         7 |                106
      208 |         8 |                105
      209 |         9 |                109
(9 rows)
```

```
DBMS_FINAL_PROJECT=# |
```

### 5.3.8 Energy

```
DBMS_FINAL_PROJECT=# SELECT *FROM mega_city_project.energy;
 energy_id | energy_type | energy_cost | energy_amountused
-----+-----+-----+-----
      302 | Gas         |      1.500 |             75.50
      303 | Solar       |      0.000 |            200.00
      304 | Solar       |      0.000 |            180.75
      305 | Wind        |      0.000 |            150.00
      306 | Electricity |      0.150 |            120.50
      307 | Hydro       |      0.000 |            200.00
      308 | Bioenergy   |      0.100 |             90.25
      309 | Gas         |      1.200 |             75.00
      301 | Electricity |      0.120 |            180.25
(9 rows)
```

```
DBMS_FINAL_PROJECT=# |
```

### 5.3.9 Consumption

```
DBMS_FINAL_PROJECT=# SELECT *FROM mega_city_project.consumption;
consumption_id | person_id | energy_id
-----+-----+-----
          401 |         1 |        301
          402 |         2 |        302
          403 |         3 |        303
          404 |         4 |        304
          405 |         5 |        306
          406 |         6 |        308
          407 |         7 |        305
          408 |         8 |        309
          409 |         9 |        307
(9 rows)

DBMS_FINAL_PROJECT=#
```

### 5.3.10 Network

```
DBMS_FINAL_PROJECT=# SELECT *FROM mega_city_project.network;
network_device_id | network_ip | c_address | person_id
-----+-----+-----+-----
      NW001       | 192.168.1.1 | Navi Mumbai |         1
      NW002       | 192.168.1.2 | Gomti Nagar |         2
      NW003       | 192.168.1.3 | MG Road     |         3
      NW004       | 192.168.1.4 | Brigade Road |         4
      NW005       | 192.168.1.5 | Juhu Beach  |         5
      NW006       | 192.168.1.6 | Hazratganj  |         6
      NW007       | 192.168.1.7 | MI Road     |         7
      NW008       | 192.168.1.8 | Ashram Road |         8
      NW009       | 192.168.1.9 | Mount Road  |         9
(9 rows)

DBMS_FINAL_PROJECT=#
```

### 5.3.11 Property

```
DBMS_FINAL_PROJECT=#
DBMS_FINAL_PROJECT=# SELECT *FROM mega_city_project.property;
property_id | property_type | property_market_value | property_name | property_address
-----+-----+-----+-----+-----
      P004  | Residential   | 2800000.000 | Sunrise Apartments | Sunset Boulevard
      P005  | Commercial   | 5500000.000 | Corporate Towers   | Tech Park Avenue
      P007  | Residential   | 3200000.000 | Green Gardens      | Rose Street
      P008  | Commercial   | 6000000.000 | Business Hub       | Corporate Avenue
      P009  | Industrial    | 9000000.000 | Tech Manufacturing | Tech Zone 3
(5 rows)

DBMS_FINAL_PROJECT=#
```



### 5.3.12 Own

```
DBMS_FINAL_PROJECT=# SELECT *FROM mega_city_project.own;
agreement_id | person_id | property_id
-----+-----+-----
A001         |          | P001
A002         |          | P002
A003         |          | P003
A004         |          | P004
A005         |          | P005
A006         |          | P006
A007         |          | P007
A008         |          | P008
A009         |          | P009
(9 rows)
```

```
DBMS_FINAL_PROJECT=# |
```

### 5.3.13 Utility

```
DBMS_FINAL_PROJECT=# SELECT * FROM mega_city_project.utility;
utility_type | person_id | utility_status | c_address
-----+-----+-----+-----
Water        |          | Active         | Residential Area
Electricity  |          | Active         | Business District
Internet     |          | Active         | Tech Zone
Water        |          | Active         | Residential Area 3
Electricity  |          | Active         | Business District 3
Internet     |          | Active         | Tech Zone 3
Water        |          | Active         | Residential Area 2
Electricity  |          | Active         | Business District 2
Internet     |          | Active         | Tech Zone 2
(9 rows)
```

```
DBMS_FINAL_PROJECT=# |
```

### 5.3.14 Safety

```
DBMS_FINAL_PROJECT=# SELECT * FROM mega_city_project.safety;
safety_incident_id | safety_incident_time | safety_incident_address | safety_incident_type | person_id
-----+-----+-----+-----+-----
S001               | 2023-05-10 15:30    | Palm Beach Road        | Fire                  | 2
S002               | 2023-07-22 09:45    | Silk Board              | Flood                 | 3
S003               | 2024-01-05 12:15    | Industrial Area         | Earthquake            | 1
S004               | 2023-05-10 16:30    | Sunset Boulevard       | Fire                  | 5
S005               | 2023-07-22 10:45    | Tech Park Avenue       | Flood                 | 6
S006               | 2024-01-05 13:30    | Industrial Zone 2       | Earthquake            | 4
S007               | 2024-02-15 14:20    | Rose Street             | Fire                  | 7
S008               | 2024-03-20 08:10    | Corporate Avenue       | Flood                 | 8
S009               | 2024-04-05 16:45    | Tech Zone 3            | Earthquake            | 9
S010               | 2024-05-10 17:30    | Main Street             | Fire                  | 7
S011               | 2024-05-10 17:30    | Main Street             | Fire                  | 7
S014               | 2024-05-10 17:30    | Main Street             | Fire                  | 7
(12 rows)
```

```
DBMS_FINAL_PROJECT=# |
```

## 6.1 QUERIES ON BASIC DBMS CONCEPT

(like, between, in , orderby, group by)

6.1.1 List persons having names starting with the letter 'R', and their corresponding email addresses.

```
DBMS_FINAL_PROJECT=#
DBMS_FINAL_PROJECT=# SELECT person_name, person_email FROM mega_city_project.person WHERE person_name LIKE 'R%';
 person_name | person_email
-----
 Rajesh Kumar | rajesh.kumar@email.com
 Raj Singh   | raj.singh@email.com
(2 rows)

DBMS_FINAL_PROJECT=#
```

6.1.2 Retrieve all travel details for travels with IDs ranging from 201 to 206.

```
DBMS_FINAL_PROJECT=#
DBMS_FINAL_PROJECT=#
DBMS_FINAL_PROJECT=# SELECT * FROM mega_city_project.travel WHERE travel_id BETWEEN 201 AND 206;
 travel_id | person_id | transport_vehical_id
-----
      201 |         1 |                101
      202 |         2 |                103
      203 |         3 |                102
      204 |         4 |                104
      205 |         5 |                108
      206 |         6 |                107
(6 rows)

DBMS_FINAL_PROJECT=#
```

6.1.3 Show the details of officers whose roles are either 'Traffic Management' or 'City Architect'.

```
DBMS_FINAL_PROJECT=#
DBMS_FINAL_PROJECT=# SELECT * FROM mega_city_project.officer WHERE officer_role IN ('Traffic Management', 'City Architect');
 officer_id | officer_role | officer_join_yr | officer_retire_yr | person_id
-----
          4 | Traffic Management |          2012 |          2024 |         4
          5 | City Architect |          2011 |          2030 |         5
(2 rows)

DBMS_FINAL_PROJECT=#
```

6.1.4 List total persons reside in each city.

```
postgres=# SELECT * FROM employee e inner join salary s on e.S_id=s.S_id where s.Salary>50000;
 e_id | a_id | e_name | e_address | e_phoneno | e_dob | s_id | s_id | salary
-----
    501 |   101 |  AJAY  |  NADIAD  | 9898160331 | 2012-09-04 | 2102 | 2102 | 60000
    503 |   102 |  SHIVA |  SURAT   | 7898763523 | 2013-07-09 | 2110 | 2110 | 100000
    504 |   103 |  STACY |  BARODA  | 7987653432 | 2013-06-12 | 2108 | 2108 | 88000
    506 |   102 |  DHYEY |  AHMEDABAD | 6733737373 | 2015-06-02 | 2105 | 2105 | 90000
    507 |   103 |  MIHIR |  NADIAD  | 6574323342 | 2015-06-09 | 2104 | 2104 | 66000
    508 |   101 |  HIL   |  SURAT   | 7865653432 | 2013-09-03 | 2107 | 2107 | 100000
    510 |   102 |  OM    |  SURAT   | 8769765346 | 2015-09-08 | 2109 | 2109 | 85000
(7 rows)
```

## 6.2 QUERIES ON JOIN, SUBQUERY, GROUPBY

### 6.2.1 Display person name and transportation type used by them.

```
SELECT p.person_name, t.transport_type
FROM mega_city_project.person p
INNER JOIN mega_city_project.travel tr
ON p.person_id = tr.person_id
INNER JOIN mega_city_project.transportation t
ON tr.transport_vehical_id = t.transport_vehical_id;
```

```
DBMS_FINAL_PROJECT=#
DBMS_FINAL_PROJECT=# SELECT p.person_name, t.transport_type FROM mega_city_project.person p INNER JOIN mega_city_project
travel tr ON p.person_id = tr.person_id INNER JOIN mega_city_project.transportation t ON tr.transport_vehical_id = t.tr
ansport_vehical_id;
person_name | transport_type
-----
Rajesh Kumar | Bus
Priya Sharma | Train
Vikram Singh | Car
Amit Patel | Metro
Neha Sharma | Electric Scooter
Raj Singh | Rickshaw
Pooja Verma | Tram
Vikram Gupta | Bicycle
Anita Kapoor | Monorail
(9 rows)

DBMS_FINAL_PROJECT=# |
```

### 6.2.2 Display IDs of officers, along with the names and the branches they administer.

```
SELECT o.officer_id, p.person_name, a.admin_branch
FROM mega_city_project.officer o
LEFT JOIN mega_city_project.person p
ON o.person_id = p.person_id LEFT JOIN
mega_city_project.admin a
ON p.person_id = a.person_id;
```

```
DBMS_FINAL_PROJECT=#
DBMS_FINAL_PROJECT=# SELECT o.officer_id, p.person_name, a.admin_branch FROM mega_city_project.officer o LEFT JOIN mega_
city_project.person p ON o.person_id = p.person_id LEFT JOIN mega_city_project.admin a ON p.person_id = a.person_id;
officer_id | person_name | admin_branch
-----
1 | Rajesh Kumar | Public Works
2 | Priya Sharma | Finance
3 | Vikram Singh | Health
4 | Amit Patel | Public Relations
5 | Neha Sharma | Education
6 | Raj Singh | Transportation
7 | Pooja Verma | Environment
8 | Vikram Gupta | Technology
9 | Anita Kapoor | Emergency Services
(9 rows)

DBMS_FINAL_PROJECT=# |
```

### 6.2.3 Retrieve the names of persons who have been involved in safety incidents.

```
SELECT person_name
FROM mega_city_project.person
WHERE person_id IN (SELECT DISTINCT person_id FROM mega_city_project.safety);
```

```
DBMS_FINAL_PROJECT=#
DBMS_FINAL_PROJECT=# SELECT person_name FROM mega_city_project.person WHERE person_id IN (SELECT DISTINCT person_id FROM
mega_city_project.safety);
person_name
-----
Rajesh Kumar
Priya Sharma
Vikram Singh
Amit Patel
Neha Sharma
Raj Singh
Pooja Verma
Vikram Gupta
Anita Kapoor
(9 rows)

DBMS_FINAL_PROJECT=# |
```

### 6.2.4 List of persons who have more than one phone number associated with them.

```
SELECT person_name
FROM mega_city_project.person p
JOIN mega_city_project.phone ph
ON p.person_id = ph.person_id
GROUP BY person_name
HAVING COUNT(*) > 1;
```

```
DBMS_FINAL_PROJECT=#
DBMS_FINAL_PROJECT=# SELECT person_name FROM mega_city_project.person p JOIN mega_city_project.phone ph ON p.person_id =
ph.person_id GROUP BY person_name HAVING COUNT(*) > 1;
person_name
-----
Neha Sharma
Pooja Verma
(2 rows)

DBMS_FINAL_PROJECT=# |
```

### 6.2.5 Find name of person who have used max energy unit.

```
SELECT person_id, person_name
FROM mega_city_project.person
WHERE person_id = (
    SELECT person_id
    FROM mega_city_project.consumption
    GROUP BY person_id
    ORDER BY SUM((SELECT energy_amountused FROM mega_city_project.energy WHERE energy_id =
consumption.energy_id)) DESC
    LIMIT 1
);
```

```
DBMS_FINAL_PROJECT=#
DBMS_FINAL_PROJECT=# SELECT person_id, person_name FROM mega_city_project.person WHERE person_id = ( SELECT person_id FR
OM mega_city_project.consumption GROUP BY person_id ORDER BY SUM((SELECT energy_amountused FROM mega_city_project.energy
WHERE energy_id = consumption.energy_id)) DESC LIMIT 1 );
 person_id | person_name
-----+-----
          3 | Vikram Singh
(1 row)

DBMS_FINAL_PROJECT=# |
```

### 6.2.6 Find person details who is citizen and uses bus for transportation.

```
SELECT p.*
FROM mega_city_project.person p
JOIN mega_city_project.citizen c ON p.person_id = c.person_id
JOIN mega_city_project.travel t ON p.person_id = t.person_id
JOIN mega_city_project.transportation tr ON t.transport_vehical_id = tr.transport_vehical_id
WHERE tr.transport_type = 'Bus';
```

```
DBMS_FINAL_PROJECT=#
DBMS_FINAL_PROJECT=# SELECT p.* FROM mega_city_project.person p JOIN mega_city_project.citizen c ON p.person_id = c.pers
on_id JOIN mega_city_project.travel t ON p.person_id = t.person_id JOIN mega_city_project.transportation tr ON t.transpo
rt_vehical_id = tr.transport_vehical_id WHERE tr.transport_type = 'Bus';
 person_id | person_name | person_password | person_email
-----+-----+-----+-----
          1 | Rajesh Kumar | pass123         | rajesh.kumar@email.com
(1 row)

DBMS_FINAL_PROJECT=# |
```

### 6.2.7 Find no. of properties own by each person

```
SELECT p.person_id, p.person_name,
COUNT(o.property_id) AS property_count
FROM mega_city_project.person p
LEFT JOIN mega_city_project.own o
ON p.person_id = o.person_id
GROUP BY p.person_id, p.person_name;
```

```
DBMS_FINAL_PROJECT=#
DBMS_FINAL_PROJECT=# SELECT p.person_id, p.person_name, COUNT(o.property_id) AS property_count FROM mega_city_project.pe
rson p LEFT JOIN mega_city_project.own o ON p.person_id = o.person_id GROUP BY p.person_id, p.person_name;
 person_id | person_name | property_count
-----+-----+-----
          5 | Neha Sharma |             1
          4 | Amit Patel  |             1
          6 | Raj Singh   |             1
          2 | Priya Sharma |            1
          7 | Pooja Verma |             1
          1 | Rajesh Kumar |             1
          8 | Vikram Gupta |             1
          9 | Anita Kapoor |             1
          3 | Vikram Singh |             1
(9 rows)

DBMS_FINAL_PROJECT=# |
```

### 6.3 PL/SQL(VIEWS):

Create a postgres view called citizen\_info that display all information about citizen.

```
CREATE OR REPLACE VIEW citizen_info AS
```

```
SELECT
```

```
c.citizen_id,  
c.c_address_state,  
c.c_address_city,  
c.c_address_street,  
p.person_id,  
p.person_name,  
p.person_email,  
ph.phone_no FROM  
mega_city_project.citizen c JOIN  
mega_city_project.person p ON c.person_id = p.person_id JOIN  
mega_city_project.phone ph ON p.person_id = ph.person_id;
```

```
SELECT * FROM citizen_info;
```

```
DBMS_FINAL_PROJECT=#  
DBMS_FINAL_PROJECT=# CREATE OR REPLACE VIEW citizen_info AS SELECT c.citizen_id, c.c_address_state, c.c_address_city, c.c_address_street, p.per  
son_id, p.person_name, p.person_email, ph.phone_no FROM mega_city_project.citizen c JOIN mega_city_project.person p ON c.person_id = p.person_  
id JOIN mega_city_project.phone ph ON p.person_id = ph.person_id;  
CREATE VIEW  
DBMS_FINAL_PROJECT=# SELECT * FROM citizen_info;  
citizen_id | c_address_state | c_address_city | c_address_street | person_id | person_name | person_email | phone_no  
-----  
581 | Maharashtra | Mumbai | Navi Mumbai | 1 | Rajesh Kumar | rajesh.kumar@email.com | 123456789  
581 | Maharashtra | Mumbai | Navi Mumbai | 1 | Rajesh Kumar | rajesh.kumar@email.com | 9876543210  
582 | Uttar Pradesh | Lucknow | Gostl Nagar | 2 | Priya Sharma | priya.sharma@email.com | 8765432189  
583 | Karnataka | Bangalore | MG Road | 3 | Vikram Singh | vikram.singh@email.com | 123456789  
583 | Karnataka | Bangalore | MG Road | 3 | Vikram Singh | vikram.singh@email.com | 7890123456  
584 | Karnataka | Bangalore | Brigade Road | 4 | Amit Patel | amit.patel@email.com | 9876543211  
585 | Maharashtra | Mumbai | Juhu Beach | 5 | Neha Sharma | neha.sharma@email.com | 123456789  
585 | Maharashtra | Mumbai | Juhu Beach | 5 | Neha Sharma | neha.sharma@email.com | 8765432188  
586 | Uttar Pradesh | Lucknow | Hazratganj | 6 | Raj Singh | raj.singh@email.com | 7890123457  
587 | Rajasthan | Jaipur | MI Road | 7 | Pooja Verma | pooja.verma@email.com | 123456789  
587 | Rajasthan | Jaipur | MI Road | 7 | Pooja Verma | pooja.verma@email.com | 123456789  
587 | Rajasthan | Jaipur | MI Road | 7 | Pooja Verma | pooja.verma@email.com | 8765432186  
588 | Gujarat | Ahmedabad | Ashram Road | 8 | Vikram Gupta | vikram.gupta@email.com | 9876543215  
589 | Tamil Nadu | Chennai | Mount Road | 9 | Anita Kapoor | anita.kapoor@email.com | 7890123454  
(14 rows)  
  
DBMS_FINAL_PROJECT=#
```

## 6.4 FUNCTION & TRIGGERS:

6.4.1 Displays the total cost (energy\_cost \* energy\_amountused) for each record using function

Function:

```
CREATE OR REPLACE FUNCTION display_total_cost_all_records()
RETURNS VOID AS $$ DECLARE
    energy_record RECORD;
    total_cost NUMERIC(10, 3);
BEGIN
    FOR energy_record IN
        SELECT * FROM mega_city_project.energy
    LOOP
        total_cost := energy_record.energy_cost *
energy_record.energy_amountused;
        RAISE NOTICE 'Total Cost for Energy %: %', energy_record.energy_type,
total_cost;
    END LOOP;
END;
$$ LANGUAGE plpgsql;
```

Check: SELECT display\_total\_cost\_all\_records();

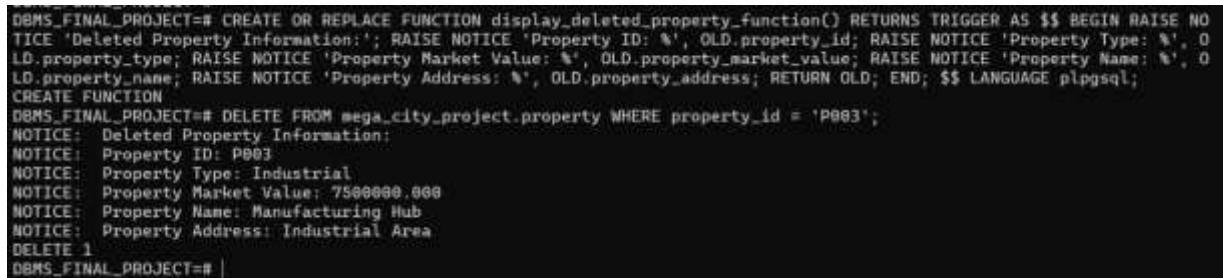
```
DBMS_FINAL_PROJECT=# CREATE OR REPLACE FUNCTION display_total_cost_all_records() RETURNS VOID AS $$ DECLARE energy_recor
d RECORD; total_cost NUMERIC(10, 3); BEGIN FOR energy_record IN SELECT * FROM mega_city_project.energy LOOP total_cost :
= energy_record.energy_cost + energy_record.energy_amountused; RAISE NOTICE 'Total Cost for Energy %: %', energy_record,
energy_type, total_cost; END LOOP; END; $$ LANGUAGE plpgsql;
CREATE FUNCTION
DBMS_FINAL_PROJECT=#
DBMS_FINAL_PROJECT=#
DBMS_FINAL_PROJECT=#
DBMS_FINAL_PROJECT=# SELECT display_total_cost_all_records();
NOTICE: Total Cost for Energy Gas: 113.250
NOTICE: Total Cost for Energy Solar: 0.000
NOTICE: Total Cost for Energy Solar: 0.000
NOTICE: Total Cost for Energy Wind: 0.000
NOTICE: Total Cost for Energy Electricity: 18.075
NOTICE: Total Cost for Energy Hydro: 0.000
NOTICE: Total Cost for Energy Bioenergy: 9.025
NOTICE: Total Cost for Energy Gas: 90.000
NOTICE: Total Cost for Energy Electricity: 21.630
display_total_cost_all_records
(1 row)
DBMS_FINAL_PROJECT=#
```

### 6.4.2 Create a function to count numbers of employees with salary between a user defined range.

Function:

```
CREATE OR REPLACE FUNCTION display_deleted_property_function()
RETURNS TRIGGER AS $$
BEGIN
    RAISE NOTICE 'Deleted Property Information:';
    RAISE NOTICE 'Property ID: %', OLD.property_id;
    RAISE NOTICE 'Property Type: %', OLD.property_type;
    RAISE NOTICE 'Property Market Value: %', OLD.property_market_value;
    RAISE NOTICE 'Property Name: %', OLD.property_name;
    RAISE NOTICE 'Property Address: %', OLD.property_address;
    RETURN OLD;
END;
$$ LANGUAGE plpgsql;
```

Check: DELETE FROM mega\_city\_project.property WHERE property\_id = 'P003';



```
DBMS_FINAL_PROJECT=# CREATE OR REPLACE FUNCTION display_deleted_property_function() RETURNS TRIGGER AS $$ BEGIN RAISE NO
TICE 'Deleted Property Information:'; RAISE NOTICE 'Property ID: %', OLD.property_id; RAISE NOTICE 'Property Type: %', O
LD.property_type; RAISE NOTICE 'Property Market Value: %', OLD.property_market_value; RAISE NOTICE 'Property Name: %', O
LD.property_name; RAISE NOTICE 'Property Address: %', OLD.property_address; RETURN OLD; END; $$ LANGUAGE plpgsql;
CREATE FUNCTION
DBMS_FINAL_PROJECT=# DELETE FROM mega_city_project.property WHERE property_id = 'P003';
NOTICE: Deleted Property Information:
NOTICE: Property ID: P003
NOTICE: Property Type: Industrial
NOTICE: Property Market Value: 7500000.000
NOTICE: Property Name: Manufacturing Hub
NOTICE: Property Address: Industrial Area
DELETE 1
DBMS_FINAL_PROJECT=# |
```

### 6.4.3 Create a trigger for calling previously created display\_deleted\_property\_function()

Trigger:

```
CREATE OR REPLACE TRIGGER display_deleted_property AFTER DELETE ON
mega_city_project.property
FOR EACH ROW
EXECUTE FUNCTION display_deleted_property_function();
```

CHECK: DELETE FROM mega\_city\_project.property WHERE property\_id = 'P006'



```

DBMS_FINAL_PROJECT=#
DBMS_FINAL_PROJECT=# CREATE OR REPLACE TRIGGER display_deleted_property AFTER DELETE ON mega_city_project.property FOR EACH ROW EXECUTE FUNCTION display_deleted_property_function();
CREATE TRIGGER
DBMS_FINAL_PROJECT=# DELETE FROM mega_city_project.property WHERE property_id = 'P006';
NOTICE: Deleted Property Information:
NOTICE: Property ID: P006
NOTICE: Property Type: Industrial
NOTICE: Property Market Value: 8000000.000
NOTICE: Property Name: Manufacturing Complex 2
NOTICE: Property Address: Industrial Zone 2
DELETE 1
DBMS_FINAL_PROJECT=# |

```

6.4.4 Create a trigger that prints “Go Firefighters!” if incident type is fire and prints “Go Rescue Team!”

#### Function:

```

CREATE OR REPLACE FUNCTION safety_trigger_function()
RETURNS TRIGGER AS $$
BEGIN
    IF TG_OP = 'INSERT' OR TG_OP = 'UPDATE' THEN
        IF NEW.safety_incident_type = 'Fire' THEN
            RAISE NOTICE 'GO FIREFIGHTERS!';
        ELSIF NEW.safety_incident_type = 'Flood' OR NEW.safety_incident_type = 'Earthquake' THEN
            RAISE NOTICE 'GO RESCUE TEAM!';
        END IF;
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

```

#### Trigger:

```

CREATE OR REPLACE TRIGGER safety_trigger
AFTER INSERT OR UPDATE ON mega_city_project.safety
FOR EACH ROW
EXECUTE FUNCTION safety_trigger_function();

```

```

DBMS_FINAL_PROJECT=#
DBMS_FINAL_PROJECT=# CREATE OR REPLACE FUNCTION safety_trigger_function() RETURNS TRIGGER AS $$ BEGIN IF TG_OP = 'INSERT' OR TG_OP = 'UPDATE' THEN IF NEW.safety_incident_type = 'Fire' THEN RAISE NOTICE 'GO FIREFIGHTERS!'; ELSIF NEW.safety_incident_type = 'Flood' OR NEW.safety_incident_type = 'Earthquake' THEN RAISE NOTICE 'GO RESCUE TEAM!'; END IF; END IF; RETURN NEW; END; $$ LANGUAGE plpgsql;
CREATE FUNCTION
DBMS_FINAL_PROJECT=#
DBMS_FINAL_PROJECT=# CREATE OR REPLACE TRIGGER safety_trigger AFTER INSERT OR UPDATE ON mega_city_project.safety FOR EACH ROW EXECUTE FUNCTION safety_trigger_function();
CREATE TRIGGER
DBMS_FINAL_PROJECT=#
DBMS_FINAL_PROJECT=# INSERT INTO mega_city_project.safety (safety_incident_id, safety_incident_time, safety_incident_address, safety_incident_type, person_id) VALUES ('S325', '2024-05-10 17:30', 'Main Street', 'Flood', 7);
NOTICE: GO RESCUE TEAM!
INSERT 0 1
DBMS_FINAL_PROJECT=# |

```

## 6.5 CURSOR:

Create a cursor retrieves the names of all persons.

```
CREATE OR REPLACE FUNCTION get_person_names() RETURNS SETOF VARCHAR AS
$$
DECLARE
    person_name VARCHAR;
    cur_person CURSOR FOR SELECT person_name FROM mega_city_project.person; BEGIN
    FOR person_rec IN cur_person LOOP
        person_name
:= person_rec.person_name;
        RETURN NEXT person_name;
    END LOOP;
    RETURN;
END;
$$ LANGUAGE plpgsql;
```

```
SELECT mega_city_project.person.person_name FROM mega_city_project.person;
```

```
DBMS_FINAL_PROJECT=# CREATE OR REPLACE FUNCTION get_person_names() RETURNS SETOF VARCHAR AS $$ DECLARE person_name VARCH
AR; cur_person CURSOR FOR SELECT person_name FROM mega_city_project.person; BEGIN FOR person_rec IN cur_person LOOP pers
on_name := person_rec.person_name; RETURN NEXT person_name; END LOOP; RETURN; END; $$ LANGUAGE plpgsql;
CREATE FUNCTION
DBMS_FINAL_PROJECT=#
DBMS_FINAL_PROJECT=# SELECT mega_city_project.person.person_name FROM mega_city_project.person;
person_name
-----
Rajesh Kumar
Priya Sharma
Vikram Singh
Amit Patel
Neha Sharma
Raj Singh
Pooja Verma
Vikram Gupta
Anita Kapoor
(9 rows)

DBMS_FINAL_PROJECT=# |
```