Database Schema

> **Source File:** generated-documents\technical-design\database-schema.md
> **Generated:** 08/07/2025 at 09:44:00
> **Generated by:** Requirements Gathering Agent - PDF Converter

DatabaseSchema

Generated by adpa-enterprise-framework-automation v3.1.6
Category: technical-design
Generated: 2025-07-05T17:02:27.917Z
Description:

Database Schema Design Document: Self-Charging Electric Vehicle (SCEV) Project

1. Database Overview:

This document outlines the database schema for the Self-Charging Electric Vehicle (SCEV) project. The database will store data related to vehicle performance, energy generation, environmental conditions, and user interactions. We will utilize a relational database management system (RDBMS) like PostgreSQL or MySQL due to their scalability and robust features. The schema is designed for optimal performance and data integrity, focusing on normalization and efficient indexing.

2. Entity Relationship Diagram (ERD):

[Insert ERD Diagram Here] *(This section would contain a visual representation of the entities and their relationships. Tools like Lucidchart, draw.io, or ERwin could be used to create this diagram.)*

3. Table Definitions:

The database will consist of the following tables:

- **Vehicles:** Stores information about individual SCEVs.
- **EnergyGeneration:** Records energy harvested from various sources for each vehicle.
- **EnvironmentalData:** Stores environmental data relevant to energy generation (e.g., sunlight, temperature).
- **BatteryData:** Tracks battery charge levels and related metrics.
- **LocationData:** Records the vehicle's location and route information.
- **Users:** Stores information about vehicle owners and drivers.
- **MaintenanceRecords:** Tracks maintenance and repair history for each vehicle.
- **SystemEvents:** Logs significant events related to the vehicle's operation and energy systems.

4. Table Details:

Table Name	Column Name	Data Type	Constraints	Description
Vehicles	vehicle_id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for each vehicle.
	vin	VARCHAR(17)	UNIQUE, NOT NULL	Vehicle Identification Number.
	model	VARCHAR(50)	NOT NULL	Vehicle model.
	manufacturing_date	DATE	NOT NULL	Manufacturing date of the vehicle.
EnergyGeneration	generation_id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for each energy generation record.
	vehicle_id	INT	FOREIGN KEY (Vehicles)	ID of the vehicle.
	timestamp	TIMESTAMP	NOT NULL	Timestamp of the energy generation event.
	source	VARCHAR(20)	NOT NULL	Energy source (solar, kinetic, thermal).
	energy_generated	DECIMAL(10, 2)	NOT NULL	Amount of energy generated (kWh).
EnvironmentalData	data_id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for each environmental data record.
	timestamp	TIMESTAMP	NOT NULL	Timestamp of the environmental

Table Name	Column Name	Data Type	Constraints	Description
				data recording.
	location	GEOGRAPHY		Location where data was recorded.
	sunlight_intensity	DECIMAL(5, 2)		Sunlight intensity (W/m^2).
	ambient_temperature	DECIMAL(5, 2)		Ambient temperature (°C).
BatteryData	battery_data_id	INT	PRIMARY KEY, AUTO_INCREMENT	Unique identifier for each battery data record.
	vehicle_id	INT	FOREIGN KEY (Vehicles)	ID of the vehicle.
	timestamp	TIMESTAMP	NOT NULL	Timestamp of the battery data recording.
	state_of_charge	DECIMAL(5, 2)	NOT NULL	Battery state of charge (%).
	battery_temperature	DECIMAL(5, 2)		Battery temperature (°C).

(This table continues for the remaining tables: LocationData, Users, MaintenanceRecords, and SystemEvents. Each table would have relevant columns with appropriate data types and constraints.)

5. Data Types:

Standard SQL data types will be used, chosen based on the nature of the data (INT for integers, VARCHAR for strings, DECIMAL for floating-point numbers, DATE/TIMESTAMP for dates and times, GEOGRAPHY for location data).

6. Indexes and Keys:

Primary keys and foreign keys will be used to enforce referential integrity. Appropriate indexes will be created on frequently queried columns to optimize query performance. Specifically, indexes will be created on `vehicle_id` in `EnergyGeneration`, `BatteryData`, and `LocationData` tables. A spatial index will be created on the `location` column in the `EnvironmentalData` table.

7. Constraints:

- **Primary Key Constraints:** Each table will have a primary key to ensure uniqueness of records.
- **Foreign Key Constraints:** Foreign keys will be used to establish relationships between tables and maintain referential integrity.
- **NOT NULL Constraints:** Appropriate NOT NULL constraints will be applied to ensure data integrity.
- **UNIQUE Constraints:** UNIQUE constraints will be used where necessary to enforce uniqueness of specific columns (e.g., VIN).
- **Check Constraints:** Check constraints can be used to validate data, ensuring that values fall within acceptable ranges (e.g., state_of_charge between 0 and 100).

8. Normalization Strategy:

The database schema will follow at least the third normal form (3NF) to minimize data redundancy and improve data integrity. Further normalization might be considered based on future requirements and data growth.

9. Performance Considerations:

- **Indexing:** Strategic indexing will be crucial for query optimization.
- **Query Optimization:** Regular review and optimization of queries will be performed to ensure efficient data retrieval.
- **Database Tuning:** Database parameters will be tuned to optimize performance based on the workload.
- **Hardware Resources:** Sufficient hardware resources (CPU, memory, storage) will be provisioned to handle the expected data volume and query load.

10. Data Migration Strategy:

A phased approach to data migration will be adopted. Initial data will be loaded from test and simulation results. As the project progresses, data from real-world testing and vehicle operation will be incrementally integrated.

11. Backup and Recovery:

Regular backups will be performed to protect against data loss. A robust recovery plan will be established to ensure quick restoration of the database in case of failure. Point-in-time recovery will be implemented to allow restoration to a specific point in time.

This schema provides a solid foundation for the SCEV project. It can be iteratively refined and extended as the project evolves and new data requirements emerge. Regular monitoring and performance testing will ensure that the database remains efficient and scalable.
