Case Study-Designing an EV Database on MySQL

Prepared by:

Nabamita De

Instructor: Dr. Junaid Qazi

Southern Alberta Institute of Technology

June 03, 2025

Nabamita De - LinkedIn

Table of Contents

Introduction:

The project outlines the design and implementation of a database system for managing EV sales, and dealership operations for a hypothetical organization.

Organization Selection:

Organization: Tee – Electrified Vehicles Division (Hypothetical)

Domain: Automotive sales and infrastructure for electric vehicles (including BEVs, PHEVs, and Hybrids).

Purpose and Goals:

Purpose: To create a database system that tracks sales, inventory, employees, and dealership data for Tee's EV business.

Goals:

- Sales Analysis: Capture and analyze sales transactions and trends over any time-period.
- Dealership Performance: Monitor and evaluate dealership performance and regional trends.
- Employee & Marketing Insights: Track sales representatives and buying trend to support targeted.

Entity Identification

In the proposed star schema, we use one fact table and four dimension tables:

Dimension Tables:

`dim_vehicle` - Electric vehicle details.

`dim_dealership` – Dealership data.

`dim_time` – Date, month, quarter, year, holiday status.

`dim_employee` – Sales representative information.

Fact Table:

`fact_sales` – Central sales table referencing all dimensions.

Attribute Specification

Table: dim_vehicle

Field	Type	Constraints
vehicle_id	INT	PRIMARY KEY
model_name	VARCHAR (50)	NOT NULL
vehicle_type	ENUM ('BEV', 'PHEV', 'Hybrid')	NOT NULL
battery_capacity	INT	NOT NULL, CHECK (battery_capacity > 0)
range_miles	INT	NOT NULL
base_price	DECIMAL(10,2)	NOT NULL
launch_year	YEAR	NOT NULL
drive_type	VARCHAR (20)	NOT NULL
charge_time	DECIMAL (4,2)	(Hours for a full charge; NULL allowed if not applicable)
colour_variety	VARCHAR (100)	Each Available colour as a separate record
VIN_number	VARCHAR(17)	NOT NULL
Engine_number	VARCHAR(17)	NOT NULL

Table: dim_dealership

Field	Type	Constraints
dealership_id	INT	PRIMARY KEY
dealership_name	VARCHAR (100)	NOT NULL
contact_email	VARCHAR (100)	UNIQUE
Phone	INT	NOT NULL
city	VARCHAR (50)	NOT NULL
province	VARCHAR (2)	NOT NULL
region	VARCHAR (20)	NOT NULL
has_fast_charging	BIT	DEFAULT FALSE
total_ev_inventory	INT	NOT NULL
display_capacity	INT	NOT NULL

Table: dim_time

Field	Type	Constraints
time_id	INT	PRIMARY KEY
sale_date	DATE	NOT NULL
month	INT	NOT NULL, CHECK (month BETWEEN 1 AND 12)
quarter	VARCHAR (5)	NOT NULL
year	INT	NOT NULL
is_holiday	BIT	DEFAULT FALSE

Table: dim_employee

Field	Type	Constraints
employee_id	INT	PRIMARY KEY, AUTO_INCREMENT
first_name	VARCHAR (50)	NOT NULL
last_name	VARCHAR (50)	NOT NULL
position	VARCHAR (50)	DEFAULT 'Sales Associate'
dealership_id	INT	FOREIGN KEY REFERENCES dim_dealership(dealership_id)

Table: fact_sales

Field	Type	Constraints
sale_id	INT	PRIMARY KEY, AUTO_INCREMENT
customer_id	INT	FOREIGN KEY REFERENCES dim_customer(customer_id)
vehicle_id	INT	FOREIGN KEY REFERENCES dim_vehicle(vehicle_id)
dealership_id	INT	FOREIGN KEY REFERENCES dim_dealership(dealership_id)
Time_period_id	INT	FOREIGN KEY REFERENCES dim_time(time_id)
employee_id	INT	(OPTIONAL—if tracking which sales rep made the sale) FOREIGN KEY REFERENCES dim_employee(employee_id)
sale_price	DECIMAL (10,2)	NOT NULL, CHECK (sale_price > 0)
tax_amount	DECIMAL (10,2)	NOT NULL
financing_used	BIT	(Indicates whether the customer used Tee financing)

Primary and Foreign Keys

- Each dimension has a unique PRIMARY KEY.
- fact_sales table includes FOREIGN KEY constraints linking to all dimension tables.
- These keys ensure referential integrity and accurate join operations

Primary Keys in Each Table

Table Name	Primary Key
dim_vehicle	vehicle_id
dim_dealership	dealership_id
dim_time	time_id
dim_employee	employee_id
fact_sales	sale_id

Foreign Keys in fact_sales Table

Foreign Key in fact_sales	Reference Table	References Column
vehicle_id	dim_vehicle	vehicle_id
dealership_id	dim_dealership	dealership_id
time_id	dim_time	time_id
employee_id	dim_employee	employee_id
Foreign Key in fact_sales	Reference Table	References Column

Establishing Entity Relationships

In our star schema design, the fact_sales table sits at the center and links directly to all

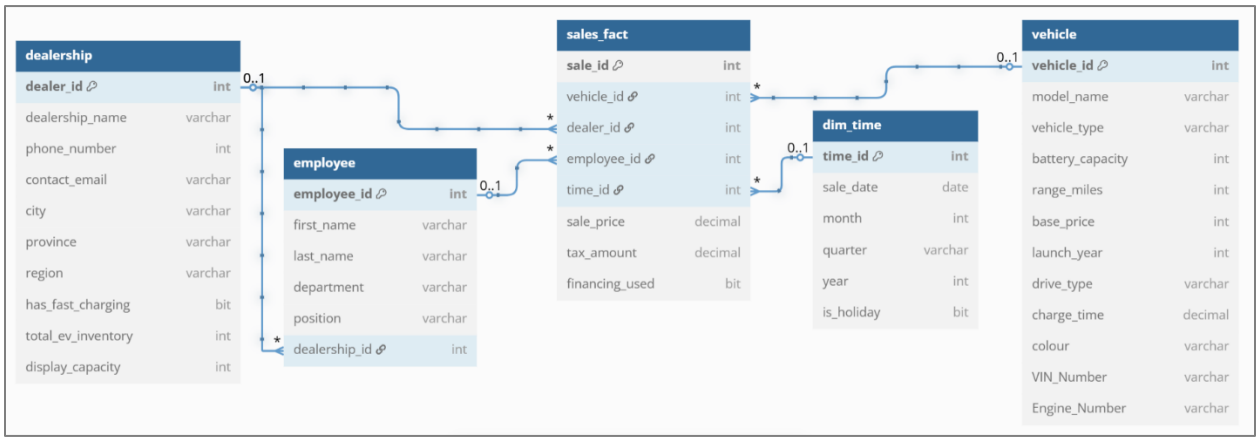
dimension tables. The relationships are one-to-many from each dimension to the fact table:

- **dim_vehicle → fact_sales:** (One to many) - Each vehicle may be associated with many sales.
- **dim_dealership → fact_sales**: (One to many) - A dealership appears in multiple sales.
- **dim_time_period → fact_sales**: (One to many) - Every sale is associated with a specific sale date (and related time attributes).
- **dim_employee → fact_sales:** (One to many) - Each sales representative may be involved in numerous sales.

Data Integrity Constraints

- Unique and non-null constraints on critical fields.
- ENUM and CHECK constraints on type, range, and price.
- Foreign key references to maintain relational consistency.

Entity-Relationship Diagram



SQL Statements

MySQL syntax is used to define all five tables with proper constraints and key definitions. (Refer to the full SQL code section.)

Sample Queries and Views

- SELECT with Filter

```sql
SELECT *
FROM dim_dealership d
WHERE d.region= 'West';
```

- SELECT with 1 JOIN and GROUP BY

```sql
SELECT
    t.year,
    t.month,
    COUNT(s.sale_id) AS total_sales,
    SUM(s.sale_price) AS total_revenue
FROM fact_sales s
JOIN dim_time t ON s.time_id = t.time_id
GROUP BY t.year, t.month;
```

- SELECT with 2 JOINS and GROUP BY

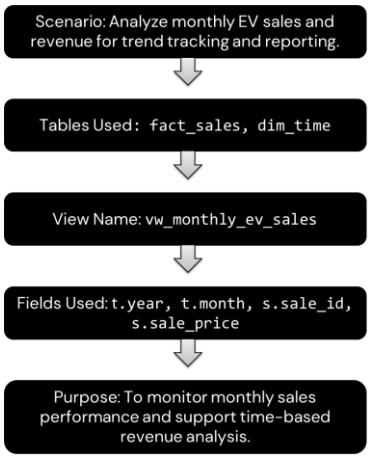
```sql
CREATE VIEW vw_top_vehicles_region AS
SELECT
    d.region,
    v.model_name,
    COUNT(s.sale_id) AS sales_count
FROM fact_sales s
JOIN dim_vehicle v ON s.vehicle_id = v.vehicle_id
JOIN dim_dealership d ON s.dealership_id = d.dealership_id
GROUP BY d.region, v.model_name;
```

Analytical views

- Monthly EV Sales

```sql
-- 1. Monthly EV Sales Report

CREATE VIEW vw_monthly_ev_sales AS
SELECT
    t.year,
    t.month,
    COUNT(s.sale_id) AS total_sales,
    SUM(s.sale_price) AS total_revenue
FROM fact_sales s
JOIN dim_time t ON s.time_id = t.time_id
GROUP BY t.year, t.month;
```



- Top Performing Dealerships

```sql
-- 2. Top-Performing Dealerships

CREATE VIEW vw_top_dealerships AS
SELECT
    d.dealership_name,
    COUNT(s.sale_id) AS num_sales,
    SUM(s.sale_price) AS revenue
FROM fact_sales s
JOIN dim_dealership d ON s.dealership_id = d.dealership_id
GROUP BY d.dealership_name;
```

- Regional vehicle trends

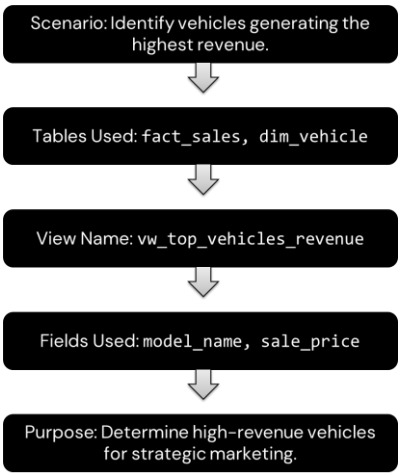
```sql
--3. Top Vehicles by Region

CREATE VIEW vw_top_vehicles_region AS
SELECT
    d.region,
    v.model_name,
    COUNT(s.sale_id) AS sales_count
FROM fact_sales s
JOIN dim_vehicle v ON s.vehicle_id = v.vehicle_id
JOIN dim_dealership d ON s.dealership_id = d.dealership_id
GROUP BY d.region, v.model_name;
```

- Revenue by vehicle

```sql
-- 4. Top Vehicles by Revenue

CREATE VIEW vw_top_vehicles_revenue AS
SELECT
    v.model_name,
    SUM(s.sale_price) AS total_revenue
FROM fact_sales s
JOIN dim_vehicle v ON s.vehicle_id = v.vehicle_id
GROUP BY v.model_name;
```



Final Notes

This design meets all database design principles, supports analytical processing, and provides scalable data management for Tee's EV division. It ensures:

- Defined purpose and business use-case alignment
- Entity and attributes clarity
- Referential integrity and normalization
- Readiness for performance monitoring and decision-making