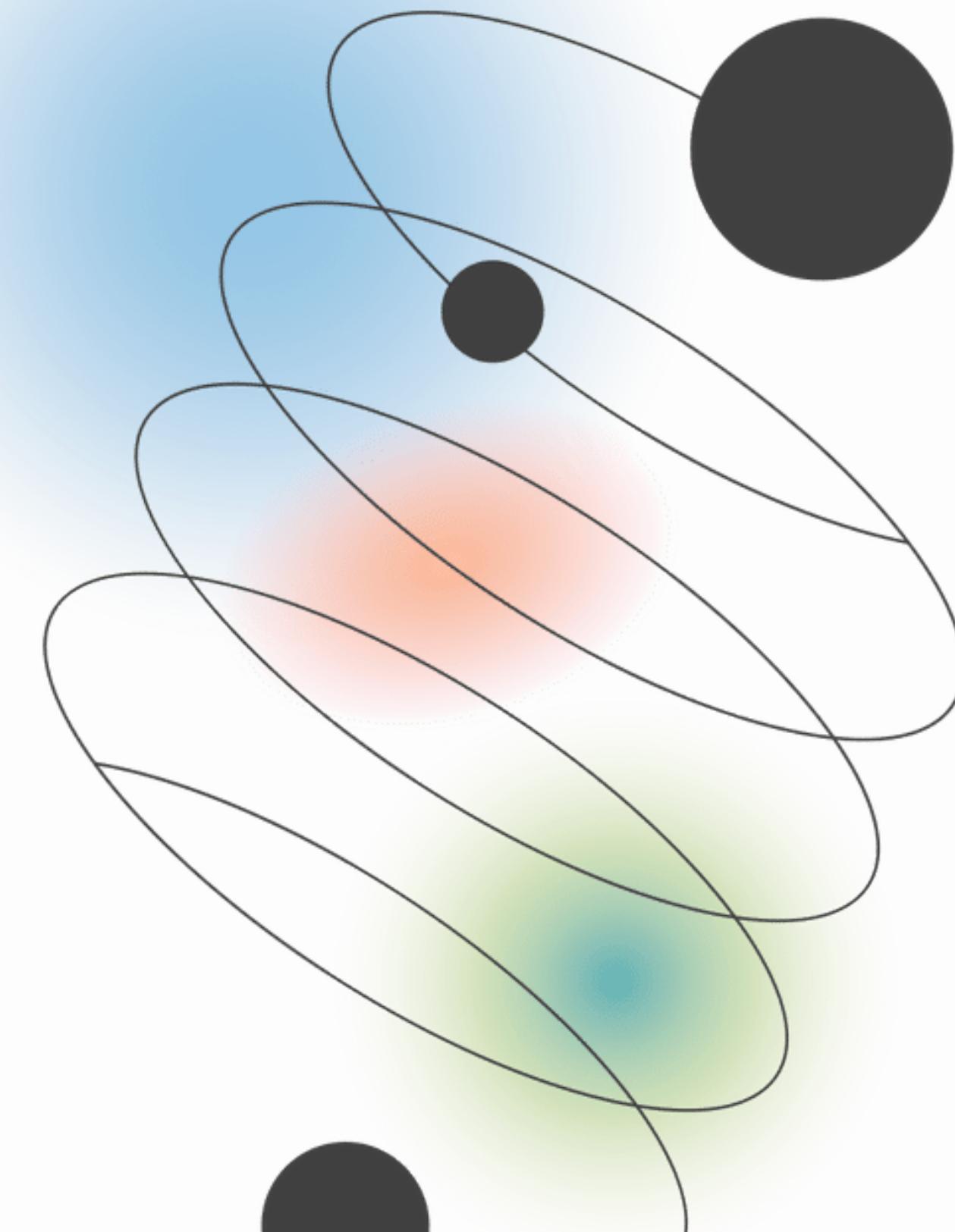


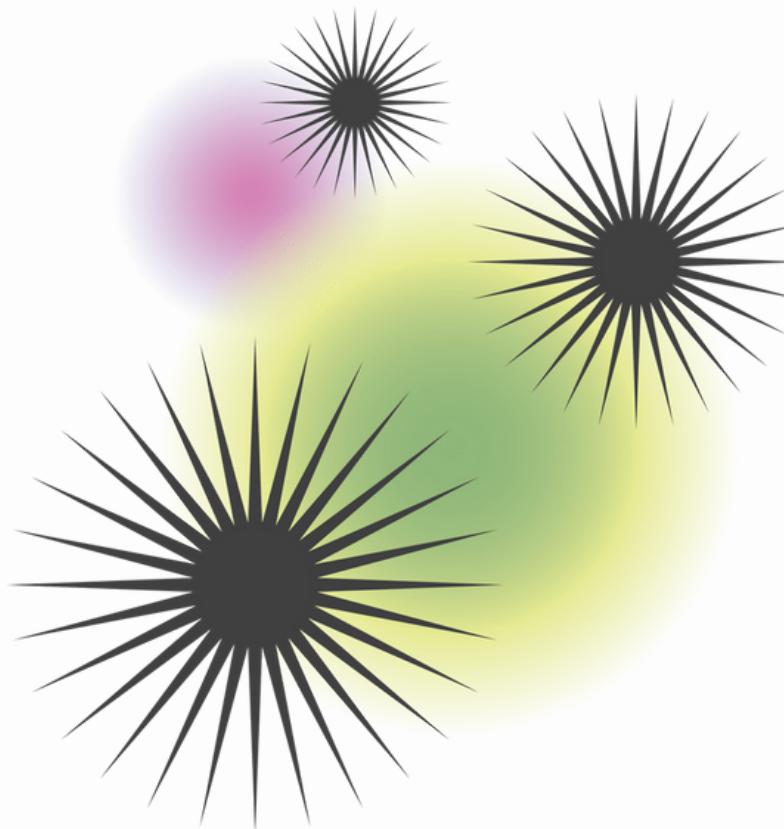
Data Mining

Project



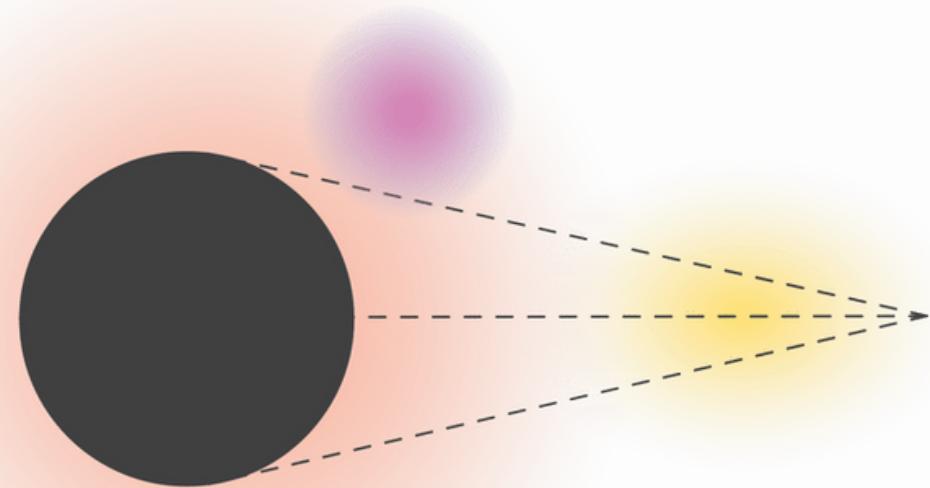
Enhancing Road Safety

A Comprehensive Analysis of Car Safety Ratings



01
Why?

03
Progress

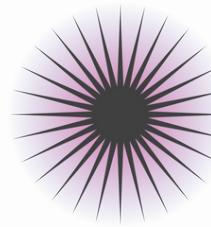
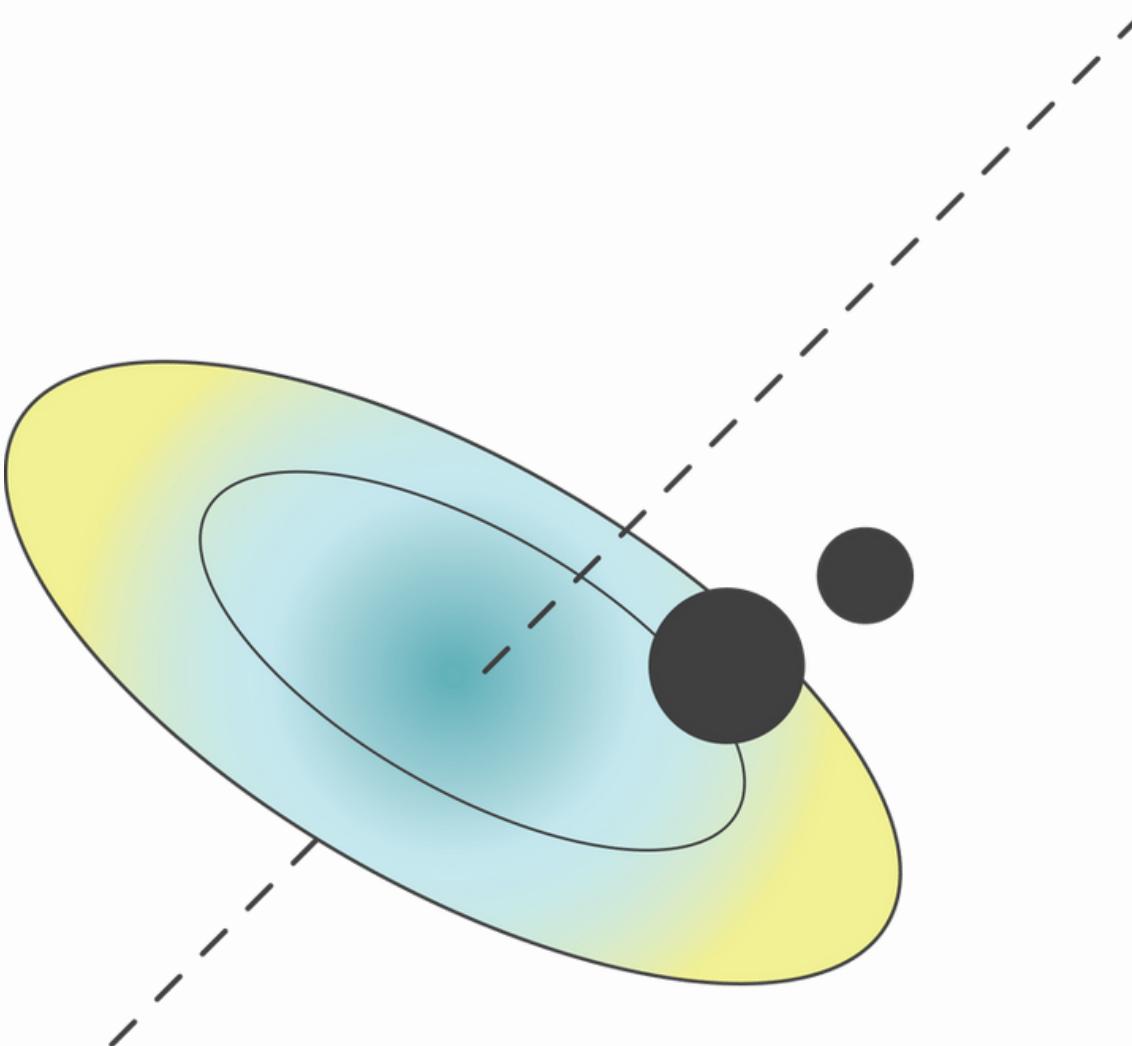


02
How

04
What's Next

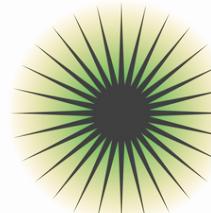
Our Goals

Two major goals
we're focusing on



Helping people make an informed decision

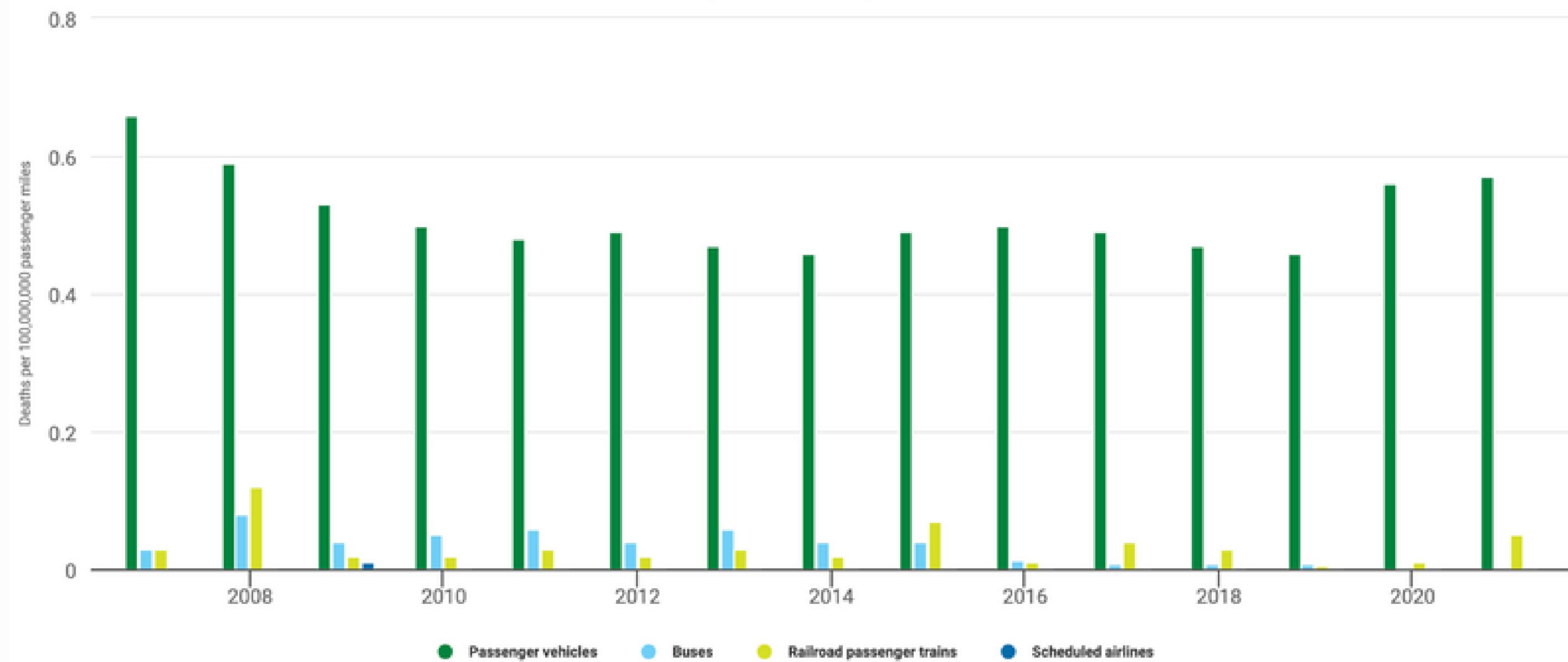
Road transport, including passenger cars, motorcycles, and trucks, tends to account for the highest number of fatalities annually due to the higher volume of vehicles and individuals using this mode of transport.



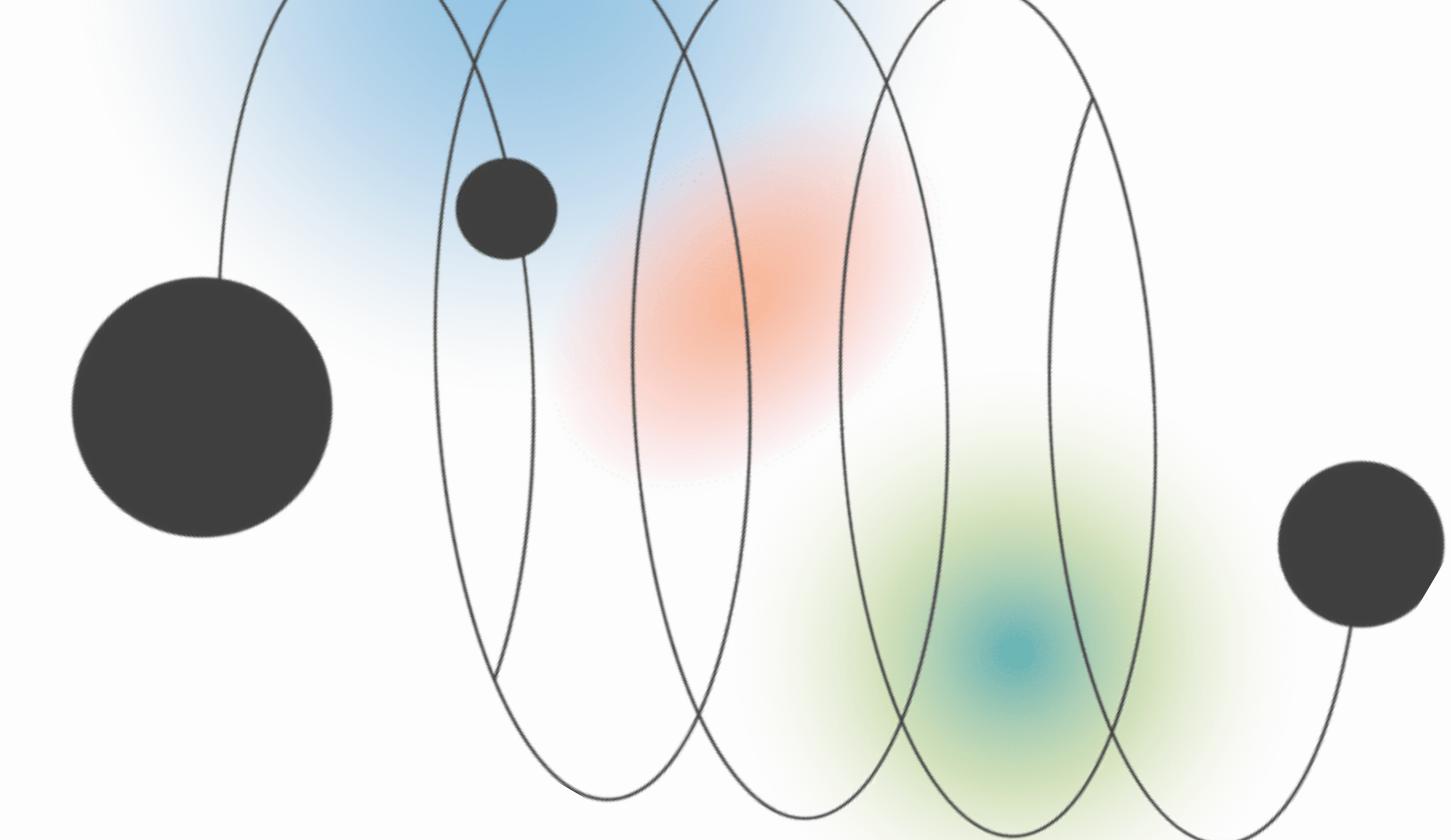
Implement in-class learnings

to leverage a detailed SQL database to analyze car safety ratings across various manufacturers, models, and years. this analysis aims to identify trends, insights, and actionable recommendations to improve vehicle safety standards.

over 94% of Deaths



5-Step Plan



01

Ideation

02

Database
Development

03

Query
Development
and Initial
Analysis

04

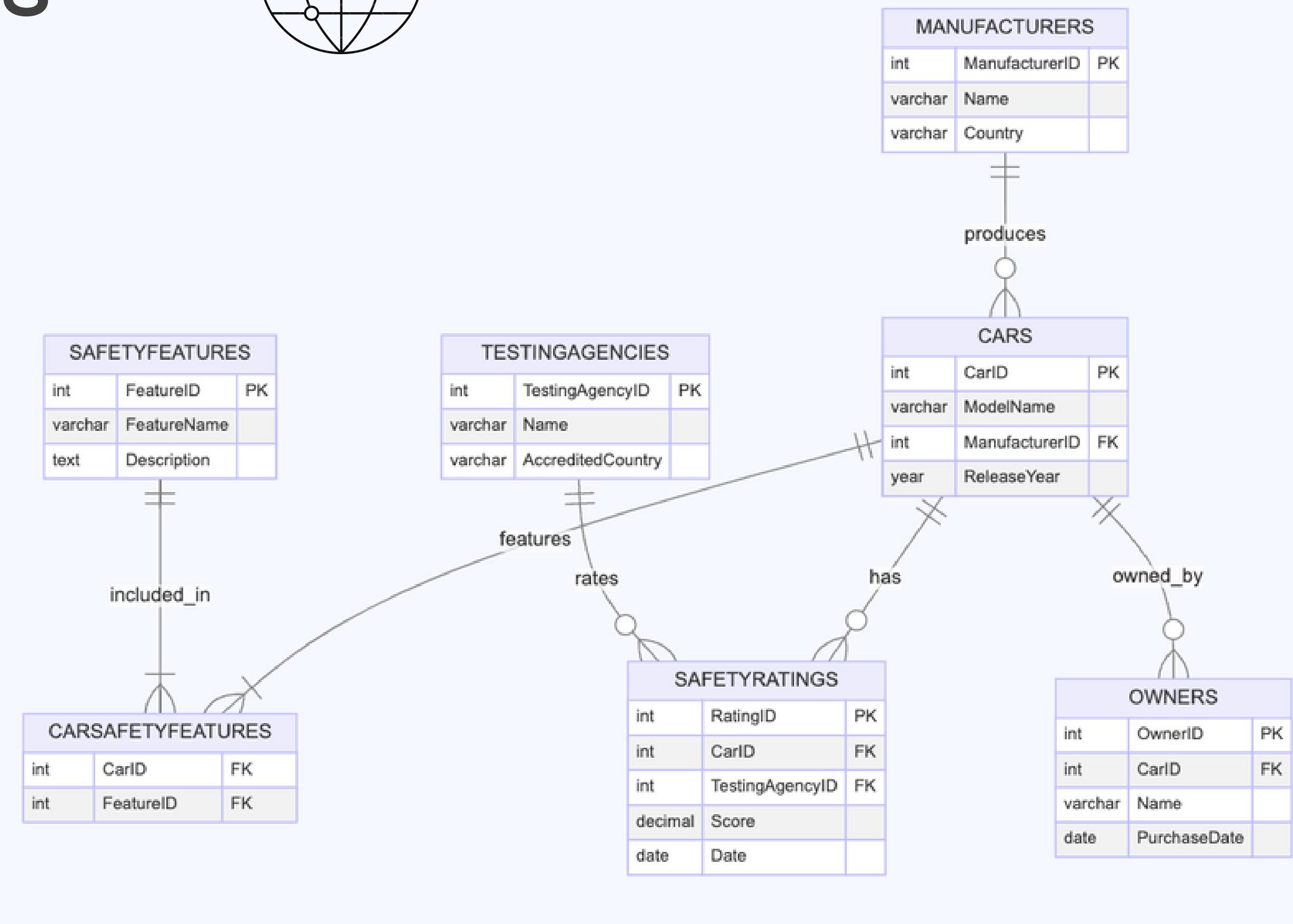
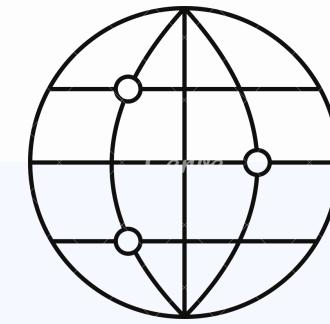
Reporting and
Visualization

05

Final
Presentation

We are here!

ER Diagram



Relations

Entities:

1. Cars: Each car model will have unique attributes.
2. Manufacturers: Companies that produce one or more car models.
3. SafetyFeatures: Safety features that can be present in car models.
4. SafetyRatings: Safety ratings provided by testing agencies.
5. TestingAgencies: Organizations that test car models and assign safety ratings.
6. Owners: Individuals or entities that own a specific car model.

Relationships:

1. One-to-One: Each car model has one (and only one) manufacturer, but for simplicity, we'll adjust this to be a one-to-many relationship because a manufacturer can produce many car models, but each car model is produced by one manufacturer.
2. One-to-Many: A manufacturer produces many car models.
3. Many-to-Many: Car models can have many safety features, and each safety feature can be in many car models. This requires a junction table.
4. Zero or One to Many (Optional One-to-Many): A car model might have zero or one safety ratings from a testing agency, but a testing agency can rate many car models.
5. One (and only one): Each car has one (and only one) safety rating from a specific testing agency.
6. Zero or Many: A car model can be owned by zero or many owners.

Triggers

```
CREATE TRIGGER LogSafetyRatingUpdate
AFTER UPDATE ON SafetyRatings
FOR EACH ROW
BEGIN
    INSERT INTO SafetyRatingChangesLog (RatingID, OldScore,
NewScore, ChangeDate)
    VALUES (OLD.RatingID, OLD.Score, NEW.Score, NOW());
END;
```

Audit Log Trigger for Safety Rating Updates

Creates a trigger to log updates to safety ratings, which could be useful for tracking changes in ratings over time.

Optimisation

```
ALTER TABLE SafetyRatings
PARTITION BY RANGE (YEAR(Date)) (
    PARTITION pBefore2020 VALUES LESS THAN (2020),
    PARTITION p2020 VALUES LESS THAN (2021),
    PARTITION p2021 VALUES LESS THAN (2022),
    PARTITION p2022 VALUES LESS THAN (2023),
    PARTITION p2023 VALUES LESS THAN (2024),
    PARTITION pAfter2023 VALUES LESS THAN MAXVALUE
);
```

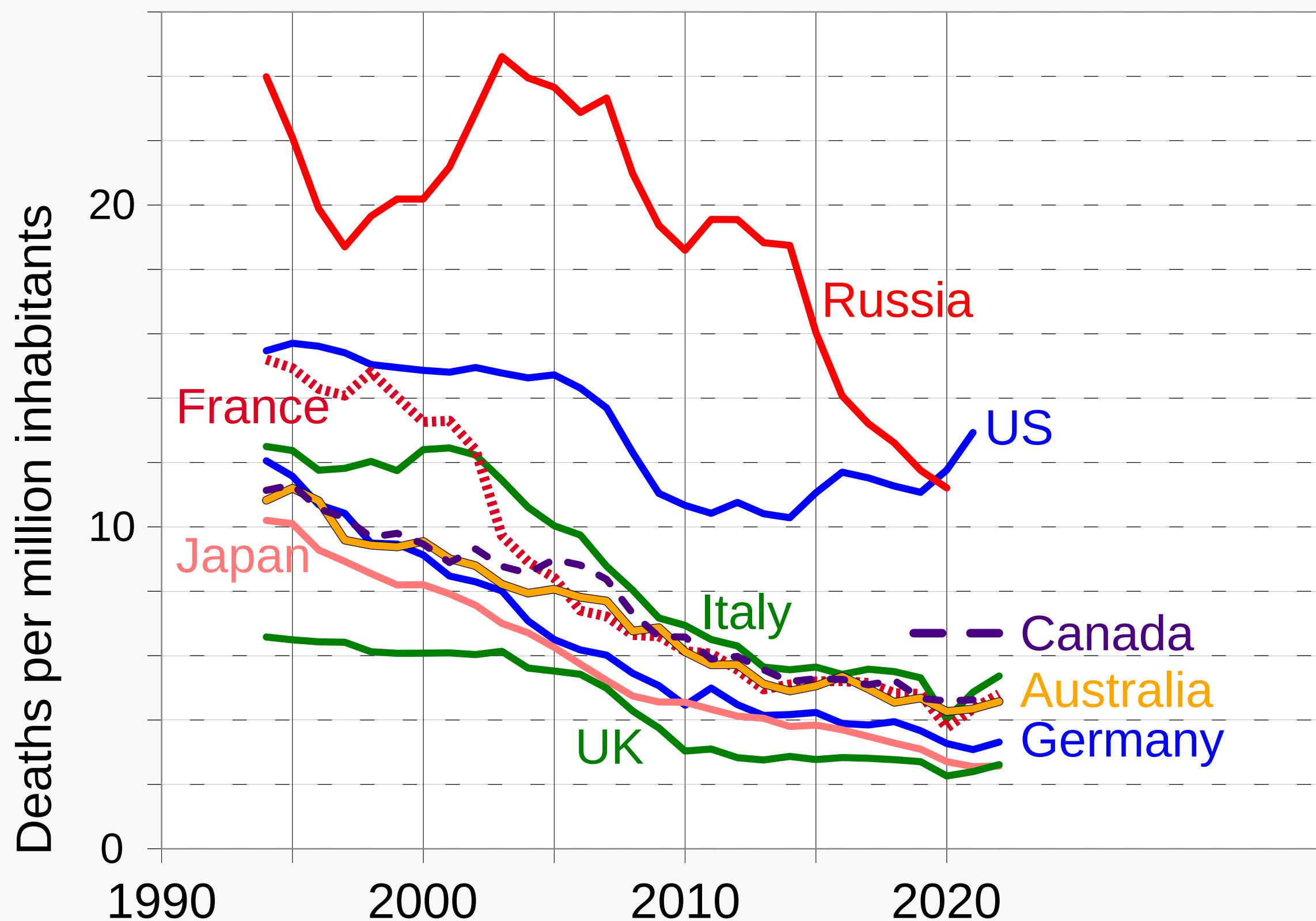
Partitioning: Large tables could benefit from partitioning. For instance, if the SafetyRatings table grows significantly, partitioning it by Date or CarID could improve query performance by limiting the number of rows scanned during a query.

Optimisation

```
-- Add indexes on foreign keys in the 'Cars' table  
CREATE INDEX idx_manufacturer_id ON Cars (ManufacturerID);  
  
-- Add indexes on foreign keys in the 'CarSafetyFeatures' table  
CREATE INDEX idx_car_id ON CarSafetyFeatures (CarID);  
CREATE INDEX idx_feature_id ON CarSafetyFeatures (FeatureID);  
  
-- Add indexes on foreign keys in the 'SafetyRatings' table  
CREATE INDEX idx_car_id_safety_ratings ON SafetyRatings (CarID);  
CREATE INDEX idx_testing_agency_id ON SafetyRatings  
(TestingAgencyID);  
  
-- Add indexes on foreign keys in the 'Owners' table  
CREATE INDEX idx_car_id_owners ON Owners (CarID);
```

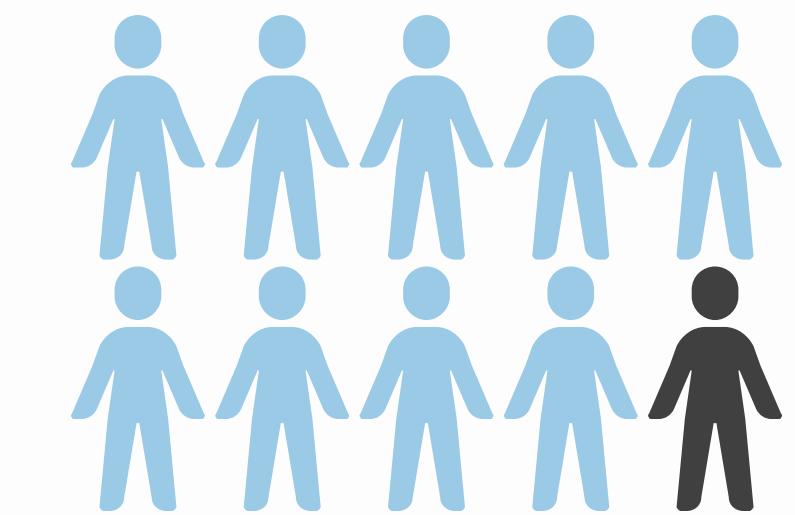
While most SQL databases automatically index primary keys, indexing foreign keys can significantly improve performance for queries involving joins.

Road accident deaths



1.19 million

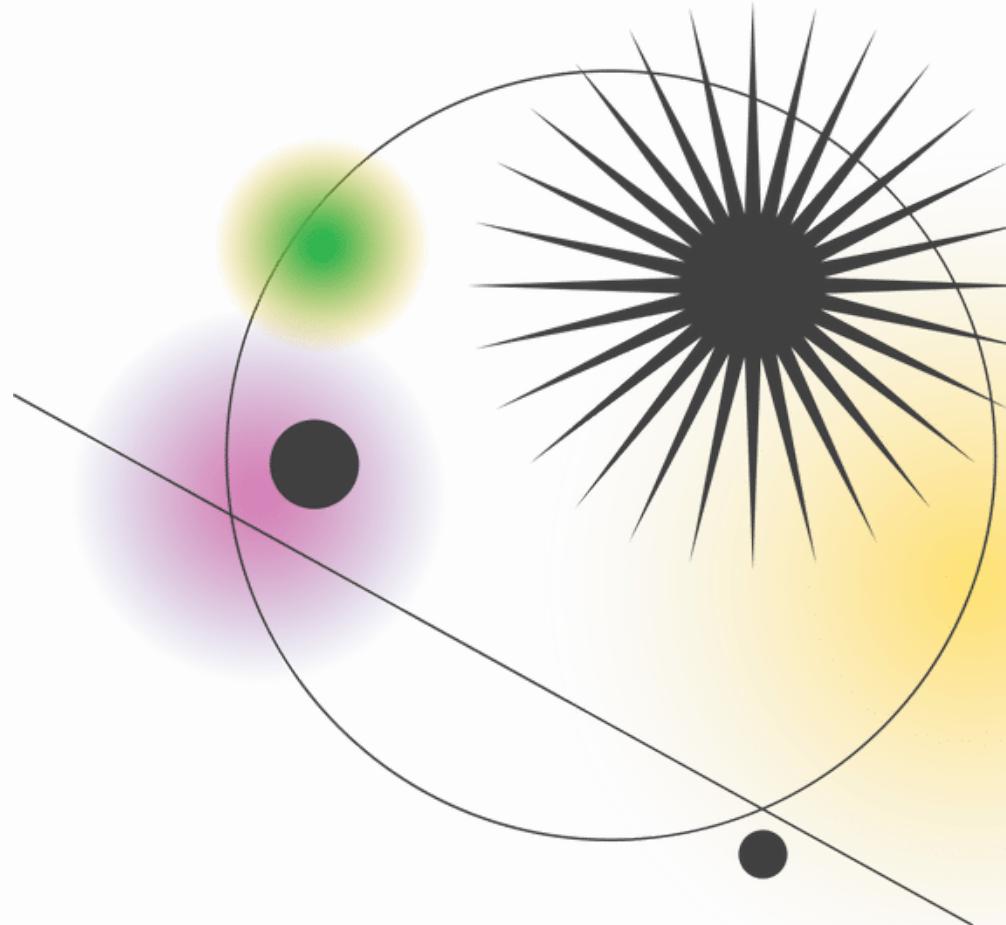
Deaths in 2023



9 out of 10

Die in road accidents

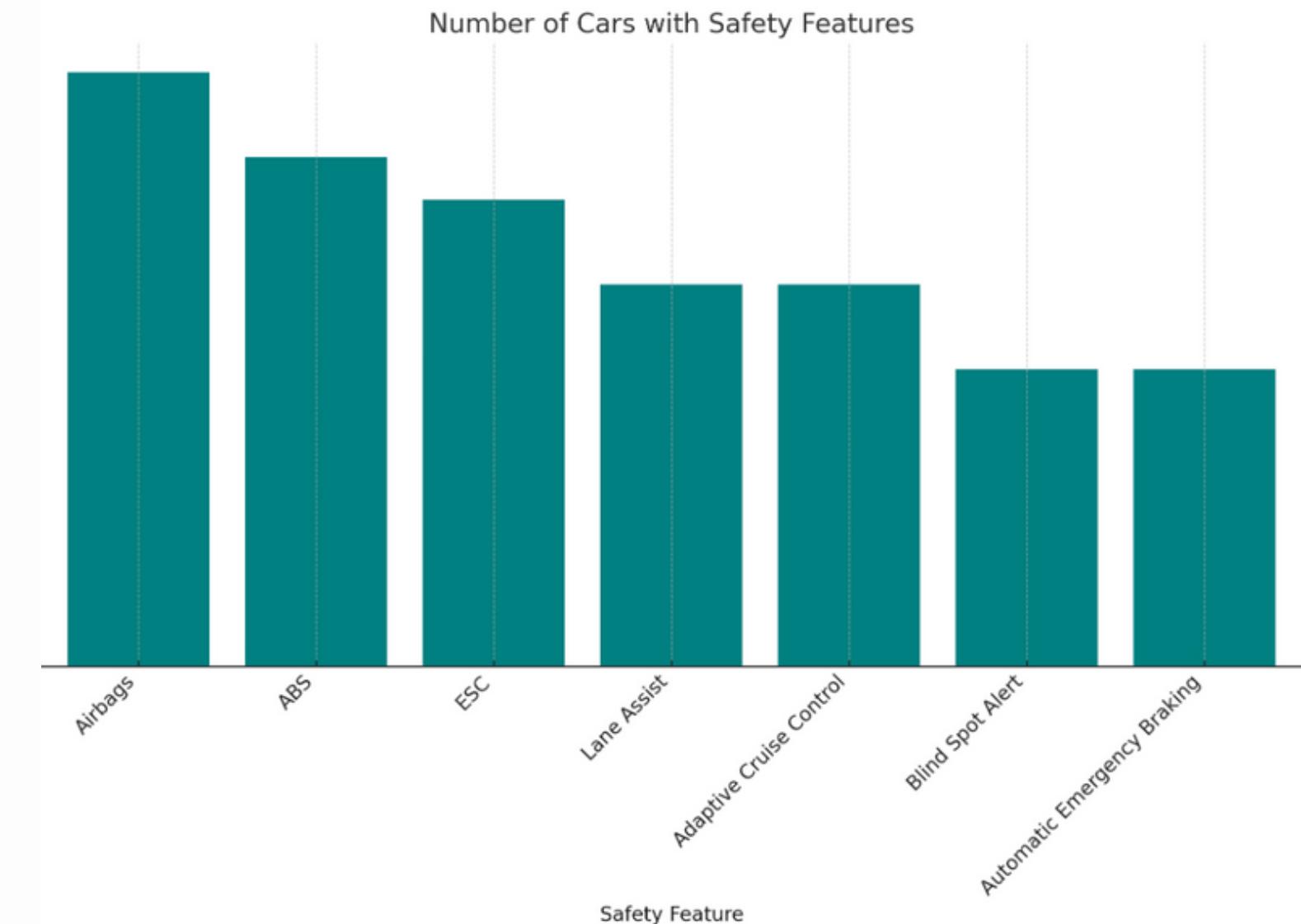
Safety Feature



1. Airbags: Inflate to cushion occupants during a crash, reducing injury risk.
2. ABS (Anti-lock Braking System): Prevents wheels from locking during braking, maintaining steering control.
3. ESC (Electronic Stability Control): Detects and reduces skidding, improving vehicle stability.
4. Lane Assist: Alerts or steers the vehicle back into the lane if it begins to drift.
5. Adaptive Cruise Control: Automatically adjusts speed to maintain a safe distance from vehicles ahead.
6. Blind Spot Alert: Warns of vehicles in the driver's blind spots during lane changes.
7. Automatic Emergency Braking (AEB): Automatically applies brakes to prevent or mitigate a collision.

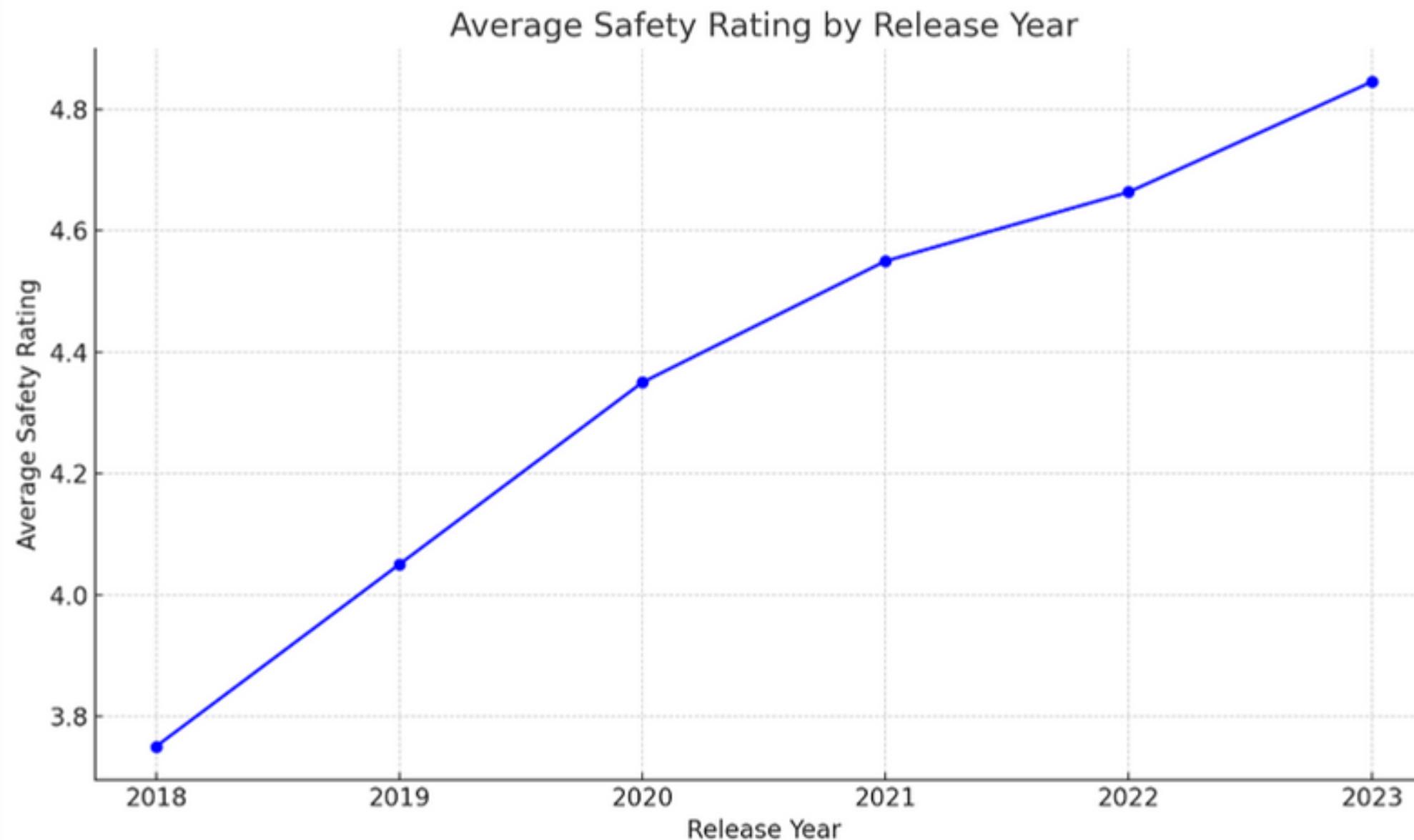
Most common safety features

```
SELECT sf.FeatureName, COUNT(csf.CarID) AS NumberOfCars  
FROM SafetyFeatures sf  
JOIN CarSafetyFeatures csf ON sf.FeatureID = csf.FeatureID  
GROUP BY sf.FeatureName  
ORDER BY NumberOfCars DESC;
```



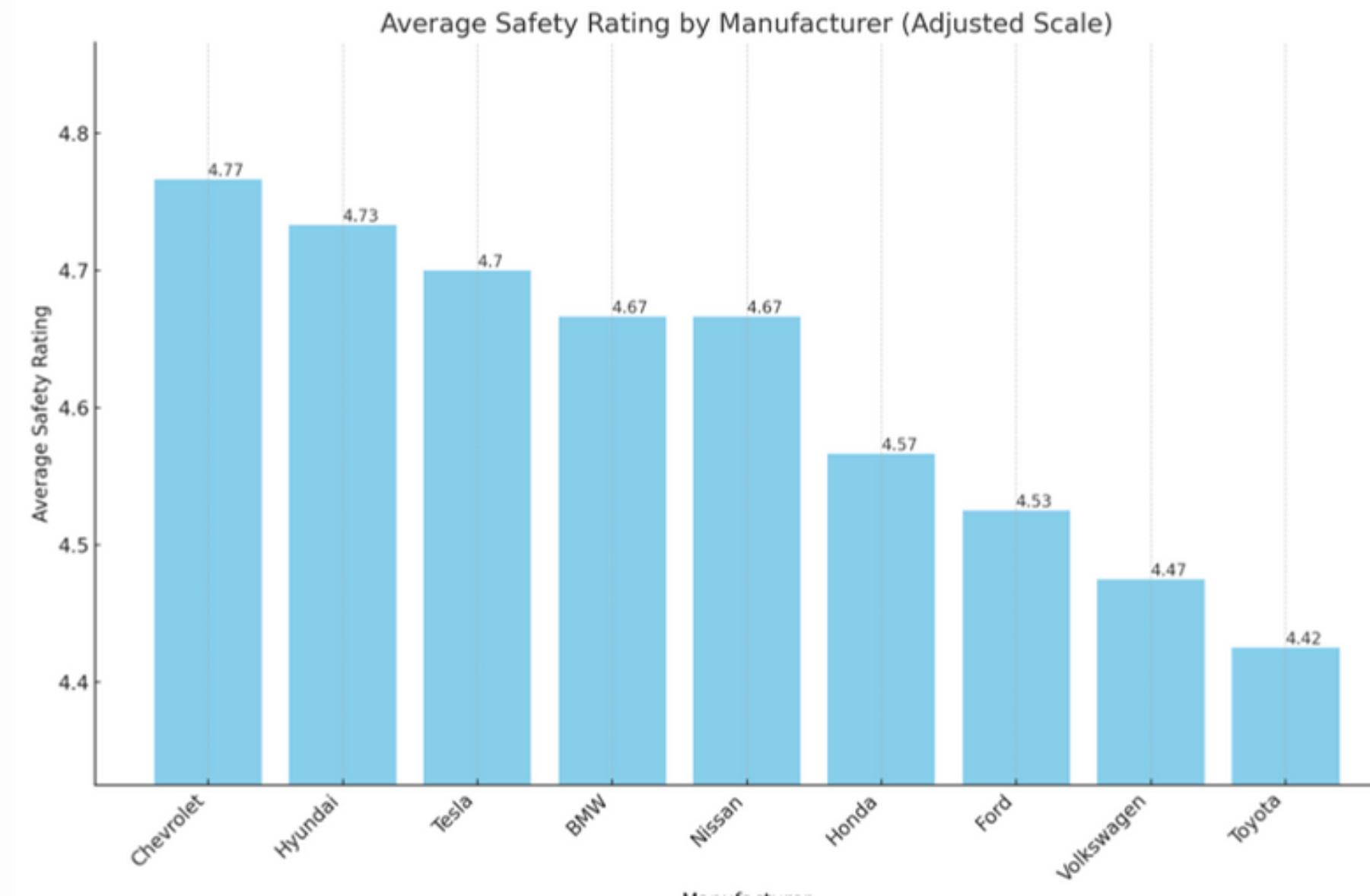
Safety Trend

```
SELECT c.ReleaseYear, AVG(r.Score) AS AvgSafetyRating  
FROM Cars c  
JOIN SafetyRatings r ON c.CarID = r.CarID  
GROUP BY c.ReleaseYear  
ORDER BY c.ReleaseYear;
```



Which manufacturer is the safest ?

```
SELECT m.Name AS Manufacturer, AVG(sr.Score) AS AverageSafetyRating  
FROM Manufacturers m  
JOIN Cars c ON m.ManufacturerID = c.ManufacturerID  
JOIN Safety Ratings sr ON c.CarID = sr.CarID  
GROUP BY m.Name  
ORDER BY AverageSafetyRating DESC;
```



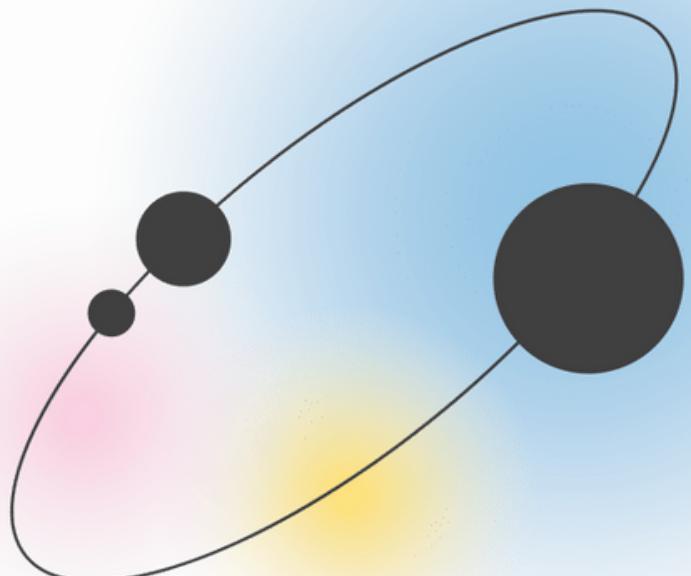
Car with most safety features

```
WITH SafetyFeaturesCount AS (
    SELECT c.CarID, c.ModelName, m.Name AS Manufacturer, COUNT(cs.FeatureID) AS FeatureCount
    FROM CarSafetyFeatures cs
    JOIN Cars c ON cs.CarID = c.CarID
    JOIN Manufacturers m ON c.ManufacturerID = m.ManufacturerID
    GROUP BY c.CarID
),
RankedCars AS (
    SELECT *,
    RANK() OVER(PARTITION BY Manufacturer ORDER BY FeatureCount DESC) AS Rank
    FROM SafetyFeaturesCount
)
SELECT ModelName, Manufacturer, FeatureCount
FROM RankedCars
WHERE Rank = 1;
```

ModelName	Manufacturer	FeatureCount
X5	BMW	4



What's Next



API Development:

Develop an API to allow third-party applications to access the database, enabling integration with car review websites, safety research organizations, and insurance companies.

Web Interface:

Develop a user-friendly web interface that allows non-technical users to easily access and interact with the database, including searching, viewing, and comparing safety ratings and features.

Thank you for attending!

Open [AI Sora](#)



Resource Page

https://road-safety.transport.ec.europa.eu/eu-road-safety-policy/priorities/safe-vehicles/archive/vehicle-safety_en

