

Database Structure:  
**Database Schema Overview**

#### 1. Claims Table

* **Columns**:
  + **ClaimID** (Primary Key)
  + ClaimDate
  + ClaimType
  + AccidentType
  + WeatherCondition
  + IncidentTime
  + PoliceReportReference
  + OutcomeID (Foreign Key to Outcome Table)
  + FraudReported
  + TotalClaimAmount
  + IncidentID (Foreign Key to Incident Table)
  + InsuranceID (Foreign Key to Insurance Table)
* **Purpose**: This table stores all the essential information about individual insurance claims, linking to incidents, drivers, vehicles, insurance policies, and outcomes. It serves as the central hub for the claims processing system.

#### 2. Incident Details Table

* **Columns**:
  + **IncidentID** (Primary Key)
  + IncidentSeverity
  + NumVehiclesInvolved
  + PropertyDamage
  + BodilyInjuries
  + Severity
  + ClaimID (Foreign Key to Claims Table)
  + VehicleID (Foreign Key to Vehicle Table)
* **Purpose**: This table captures detailed information about each incident associated with a claim, including the severity, number of vehicles involved, and any damage or injuries sustained.

#### 3. Driver Information Table

* **Columns**:
  + **DriverID** (Primary Key)
  + DriverGender
  + DriverAge
  + DriverPolicyNo
  + DriverExperience
  + LicenceType
  + DUI (Driving Under Influence)
  + AccidentHistory
  + CreditScore
  + CriminalHistory
  + EducationLevel
  + Time as Customer
  + ClaimID (Foreign Key to Claims Table)
  + InsuranceID (Foreign Key to Insurance Table)
* **Purpose**: This table holds personal and historical information about the drivers involved in claims, linking them to their respective claims and insurance policies.

#### 4. Vehicle Information Table

* **Columns**:
  + **VehicleID** (Primary Key)
  + VehicleType
  + VehicleYear
  + VehicleModel
  + AtFault
  + DamageAmount
  + IncidentID (Foreign Key to Incident Table)
  + ClaimID (Foreign Key to Claims Table)
  + DriverID (Foreign Key to Driver Table)
* **Purpose**: This table stores details about the vehicles involved in incidents, including their type, model, year, and any damage sustained, along with their relationship to incidents, claims, and drivers.

#### 5. Insurance Information Table

* **Columns**:
  + **InsuranceID** (Primary Key)
  + InsurancePremium
  + InsuranceAccess
  + DriverID (Foreign Key to Driver Table)
  + ClaimID (Foreign Key to Claims Table)
* **Purpose**: This table contains information about the insurance policies tied to each claim, linking them to the respective drivers and claims.

#### 6. Outcome Information Table

* **Columns**:
  + **OutcomeID** (Primary Key)
  + Outcome
  + ClaimID (Foreign Key to Claims Table)
* **Purpose**: This table records the outcomes of each claim, indicating whether the claim was successful, rejected, or flagged as fraudulent.

### Relationships and Keys

* **Primary Keys (PK)**: Each table has a primary key (e.g., ClaimID, IncidentID) that uniquely identifies each record within that table.
* **Foreign Keys (FK)**: Foreign keys establish relationships between tables, allowing you to link related data across different tables. For instance, ClaimID in the Incident Details Table connects an incident to its corresponding claim in the Claims Table.
* **Relationships**:
  + **Claims** are linked to **Incidents**, **Drivers**, **Vehicles**, **Insurance**, and **Outcomes**.
  + **Incidents** are connected to **Vehicles** and **Drivers**.
  + **Drivers** are connected to **Insurance**.
  + **Vehicles** are connected to **Drivers**.

### Benefits of the Schema

1. **Normalized Data Structure**: The database schema is well-organized and normalized, reducing data redundancy and ensuring consistency across all tables.
2. **Scalability**: The schema is designed to scale, allowing for the easy addition of more claims, incidents, drivers, vehicles, and insurance records as needed.
3. **Clear Relationships**: Well-defined relationships between entities make it easy to query and analyze data across different dimensions, such as linking claims to their outcomes, or incidents to the vehicles involved.
4. **Efficiency**: By structuring the data into separate, related tables, the schema supports efficient data retrieval and management, which is crucial for processing and analyzing large volumes of claims data.
5. **Flexibility**: The schema is flexible enough to accommodate additional fields or tables as the system evolves, allowing for future expansion or integration with other data sources.

Fraud Detection Database Layout on Google Sheets

<https://docs.google.com/spreadsheets/d/1soajJ-KTmRZbMKtFeCQ7Bqh1EptbwEbWPht6xrgkBs4/edit?usp=sharing>

**Explanation:**

1. **Consolidated Data Sheet**:
   * Contains the essential columns that integrate information from all other tables, minimizing repetition.
   * Key identifiers like ClaimID, IncidentID, and DriverID are not repeated where not necessary.
2. **Claims Table**:
   * Focuses on claims-related information and includes references (IncidentID, InsuranceID) to other tables without unnecessary duplication.
3. **Incident Details Table**:
   * Includes detailed information about the incident, linked to claims and vehicles through foreign keys.
4. **Driver Information Table**:
   * Contains driver-related data, with connections to their claims and insurance information.
5. **Vehicle Information Table**:
   * Captures vehicle-specific details and their involvement in incidents and claims.
6. **Insurance Information Table**:
   * Covers insurance policy details, linked to drivers and claims.
7. **Outcome Information Table**:
   * Stores the outcome of each claim, directly tied to the ClaimID.

### Relationships Overview:

* **Driver** is linked to **Insurance** and **Claims** via DriverID and InsuranceID.
* **Vehicle** is linked to **Incident** and **Claims** via VehicleID and IncidentID.
* **Incident** is linked to **Claims** via IncidentID.
* **Insurance** is linked to **Driver** and **Claims** via InsuranceID.
* **Outcome** is directly linked to **Claims** via OutcomeID.