INFO260 Course Project: Submission Two

Matthew D. Sherman

Department of Accounting and Information Systems, University of Canterbury

INFO260: Data Management

October 23, 2023

Table of Contents

Submission One (Grade: 96%)

Note: I have copied and pasted everything my first submission here and have not made any edits apart from some formatting changes to fit with the format of this submission.

Purpose of Data Management

Analysing the Ambulance and Paramedic Dispatch System
The case presents the Emergency Medical Services (EMS), who provide fast and effective medical response to a wide array of medical emergencies in New Zealand. Despite the skills and capabilities of the EMS team, the dispatch process is currently inefficient. This can occasionally result in delays in responding to critical emergencies. The current dispatch capabilities can also lead to poor resource allocation, hampering the delivery efficiency. Hence, the EMS board members are interested in an ambulance dispatch system which is efficient and accurate. The EMS envision a dispatch system which enhances dispatch response times and the outcomes of patients by utilising data-driven insights. The EMS are also interested in an improved data management plant which can enhance dispatch efficiency, optimize ambulance allocation, and evaluate performance effectively.

Business requirements and data requirements of the system.
The EMS department has identified two main business requirements of the EMS system which are dispatching efficiency and dispatching accuracy. By meeting these requirements, the EMS will be able to significantly improve their service and help save lives in New Zealand.

- Dispatching efficiency
  The current dispatch process is inefficient and can sometimes result in delays. The EMS would like to improve this as it would mean that the EMS can respond to emergencies faster. The EMS department aims to do this by using data management to quickly identify the nearest available ambulance which has the appropriate medical facilities and paramedics.
- Dispatching Accuracy
  Accuracy in the dispatch process is very important for the EMS. This is because the EMS has limited resources, and so must allocate them effectively. The current dispatch system sometimes leads to suboptimal resource allocation, resulting in care which is inefficient. Dispatching accuracy can be improved by the data system prioritising incident based on medical conditions. This is so that the most appropriate ambulance and paramedics are sent to the scene.

Overall, the business requirements are very straightforward and logical. If implemented, should make a world of difference is the service provided to New Zealanders by the EMS.

The data requirements of the EMS system include:

- Efficient and effective capturing, storing, and analysing of critical incident data.
- Comprehensive analysis of the data
  - Identifying patterns
  - Assessing response times
  - Evaluating dispatch accuracy
- Establishing a data warehouse to effectively store/manage large amounts of both historical and real-time data.

- Implementing an Extract, Transform, Load (ETL) plan to extract data from different sources, convert it into a consistent format, and then transfer it into the data warehouse.
- A sophisticated data environment the supports:
    - Advanced analytics
    - Reporting
    - Data mining

The importance of effective data management for the Ambulance and Paramedic Dispatch System.
Effective data management is very important for the Ambulance and Paramedic Dispatch System for many reasons including:

- **Faster response times**
  The response times can be reduced by rapidly identifying the nearest ambulance which has the appropriate medical facilities and qualified paramedics.
- **Better dispatch accuracy**
  Dispatch accuracy can be improved by prioritizing incidents based on the given medical condition. Then, the most appropriate ambulance and paramedics can be sent to the incident as soon as possible.
- **Optimised resource allocation**
  By sending the most appropriate ambulances and paramedics to each incident, the limited resources of the EMS can be utilised optimally.
- **Storing and analysis of data to make data-driven decisions**
  By using effective data management, the EMS can effectively store important data and then analyse this data to gain insights. Then these insights can be used to make data-driven decisions that will benefit the EMS.
- **Better service**
  The EMS having effective data management means that they can respond to emergencies faster and with the right resources. This would greatly increase their quality of service in helping patients.
- **Increased trust and public image**
  By providing fast and accurate care, New Zealand citizens will trust the EMS more in delivering medical care and, in turn, the public image of the EMS will heighten.
- **Saving lives**
  Overall, effective data management is very important for the EMS as the stakes are incredibly high and there are potential lives on the line. By implementing effective data management and gaining the above benefits, the EMS can continue to help save more lives.

Database Design

Database Design
The EMS database in its current state is not meeting the needs and requirements that the EMS department are interested in. The current database is a single table and, along with not meeting the requirements, also suffers from issues such as data redundancy, a lack of normalisation, and poor data integrity to name a few. Hence why it is imperative that a new database design is created.

To address the dispatching efficiency business requirement, the EMS aims to identify the nearest available ambulance (to the incident) which has the appropriate medical facilities and paramedics. To achieve this, we will need to know the location of the incident, the nature of the incident (to receive the appropriate medical facilities and paramedics), the location of the available ambulances and their vehicle type, and the available paramedics and their qualification.

To address the dispatching accuracy business requirement, the EMS wants the system to prioritise incidents based on medical conditions so that the appropriate ambulance type and qualified reach the scene quickly. To do this, we can assign a dispatch priority to each incident based off the details provided in the call to the EMS. This way, incidents that have more severe medical conditions (as described in the phone call) receive a higher dispatch priority and, in turn, these incidents are prioritised in the dispatch process.

The data requirements as given by the EMS department will also need to be considered in the database design. However, many of the given data requirements are not relevant in the database design and will be addressed later on. The main data requirement for the database is that it stores the necessary and relevant data such as patient information, incident details, clinical impressions, etc.

It is also important that the database is designed well so that it is normalised, does not have data redundancy, has good data integrity, etc. so that the database is efficient and supports the EMS' needs effectively.

<u>Designing the database schema</u>
Initial tables:

- Patient
  There needs to be a patient table as each incident deals with one patient, and the EMS aims to care for patients. There can only be one patient for each incident, but a given patient can be the patient in many different incidents.
- Incident
  Incidents could be considered the core data point as an incident happening is what causes the EMS response.
- Clinical Assessment
  This is about storing clinical data gathered by the paramedic once they've arrived to the incident relating to the patient in the incident.
- Clinical Impression
  This is related to clinical assessment and is about the clinical impressions gathered by the paramedics about the patient. The impressions are in order from most to least prominent.
- Glasgow Coma Scale
  This also relates to clinical assessment and is about evaluating the consciousness of the patient.
- Dispatch Information
  This is relating to the business requirements and is about the information used to decide the dispatch to the incident.
- Dispatch
  This is about the dispatch (vehicle type, paramedics, etc.) to the incident.
- Ambulance
  This is about the ambulance that can be dispatched to the incident.

- Paramedic
  This is about the paramedics which can be dispatched to the incident.

These are the initial tables as more will be created once some more are normalised.

<u>Attributes and primary keys for each table:</u>

Patient

- PatientID
- Date of birth
- Gender

PatientID is needed to uniquely identify the patients and is the primary key.
Date of birth is used to identify the patient's age. Previously, there were three attributes – age, agetype, and agecategory – but having three attributes to represent age seemed redundant. Date of birth can achieve the same purpose in one attribute. Needed for general information about the patient (can be used later on in analysis and categorisation).
Gender is used to identify the patient's gender. Needed for general information about the patient (can be used later on in analysis and categorisation).
The database could possibly use the DOB and gender to assist in dispatch prioritisation and choosing the right ambulance/paramedics, but this is not a requirement and so will not be implemented as to keep the database simple.

Incident

- IncidentID
- Call_district
- DateOfCall
- DateOnset
- Location_type
- CAD_Time_PhonePickUp
- IsSTEMI
- IsOHCA
- CPRPerformed

IncidentID is needed to uniquely identify the incident and is the primary key.
Call_district is the district from which the call originates. Needed so that the dispatch can know where to go to.
DateOfCall is when the call is made. Needed for general incident information.
DateOnset is when the patient's symptoms started. Needed for general incident information and data storing.
Location_type is about the location type (urban/rural) of the incident location. Needed for general incident information and data storing.
CAD_Time_PhonePickUp is about when the EMS picked up the phone call. Needed for general incident information and data storing.

Clinical Assessment

- ClinicalAssessmentID
- IsSTEMI
- IsOHCA
- CPRPerformed

ClinicalAssessmentID is for uniquely identifying the clinical assessment table and is the primary key.

IsSTEMI**,** IsOHCA**,** and CPRPerformed are important clinical assessment measures which is why they are included as attributes.

Clinical Impression

- ClinicalImpressionID
- ImpressionOrder
- ClinicalImpression

ClinicalImpressionID is for uniquely identifying the clinical impression table and is the primary key.
ClinicalImpression and ImpressionOrder are used to list clinical impressions. The impression order is needed as it is important that the impressions are ordered from primary impression onwards.

Glasgow Coma Scale

- GCSID
- GCS_Initial
- GCS_Initial_Time
- GCS_Final
- GCS_Final_Time

GCSID is for uniquely identifying the GCS table and is the primary key.
GCS_Initial is the first GCS measure recorded when the paramedics arrive,
GCS_Initial_Time is the time that this is recorded.
GCS_Final is the first GCS measure recorded when the paramedics arrive, GCS_Final_Time is the time that this is recorded.

Dispatch Information

- DispatchInformationID
- CAD_Response_Area
- CAD_Triage_Priority

DispatchInformationID is for uniquely identifying the Dispatch Information **t**able and is the primary key.
CAD_Triage_Priority is to prioritise the incidents by severity based on the phone call (which gives CAD_Triage_Priority) and meet the dispatch accuracy business requirement of sorting incidents by the medical condition severity.
CAD_Response_Area is the location of the incident. This is needed so that the closest appropriate ambulance with the qualified paramedics can be identified.
There is also CAD_Service_Area which is separated into a different table as to comply with normalisation.

Service Area

- CAD_Response_area
- CAD_Service_area

CAD_Response_area is for uniquely identifying the Service Area table and is the primary key. This is because service area can be identified by the response area.
CAD_Service_area is the area type (urban/rural/remote). Is needed for the EMS to respond accordingly.

Dispatch

- DispatchID
- CAD_Time_Assigned
- CAD_Time_ArrivedAtScene
- CAD_Time_Depart_Scene
- CAD_Time_Arrive_Destination
- FinalDestination

DispatchID is for uniquely identifying the dispatch table and is the primary key.
CAD_Time_Assigned, CAD_Time_ArrivedAtScene, CAD_Time_Depart_Scene, CAD_Time_Arrive_Destination are the relevant times throughout the dispatch process.
FinalDestination is the final destination that the dispatch takes the patient to.
There is also FinalDestination_IsPCIHospital which is whether a hospital has a cardiac treatment facility or not. This is given a new table to comply with normalisation.

Cardiac Treatment Facility

- FinalDestination
- FinalDestination_IsPCIHospital

FinalDestination is for uniquely identifying the Cardiac Treatment Facility table and is the primary key. This is because FinalDestination_IsPCIHospital can be identified by the final destination.

Ambulance

- AmbulanceID
- VehicleType
- AmbulanceLocation
- AmbulanceAvailability

AmbulanceID is for uniquely identifying the ambulance table and is the primary key.
VehicleType is the type of vehicle the ambulance is and the medical capabilities it provides. The database aims to dispatch the adequate vehicle to the appropriate incident given the incident's medical severity.
AmbulanceLocation is used to know the location of the ambulance so that ambulances that are closer can be assigned to the incident.
AmbulanceAvailability is used to know if an ambulance is available, so can be dispatched to an incident.

Paramedic

- ParamedicID
- ParamedicQualification
- ParamedicAvailability

ParamedicID is for uniquely identifying the ambulance table and is the primary key.
ParamedicQualification is used to get the qualification of the paramedic. This way, the paramedics with the suitable qualifications can be dispatched to the incident (fitting with the requirements).
ParamedicAvailability is used to know if a paramedic is available to be dispatched or not.

Relationships:

There can be one patient per incident, but a patient may be involved in many different incidents. Hence, there is a one-to-many relationship between Patient and Incident. It is mandatory for both sides as for a patient to exist, they must have been associated with an incident, and for an incident to exist, there must be a patient.

There is a one-to-one relationship between incident and dispatch information as an incident can have one instance of dispatch information and vice versa. It is mandatory for both sides as an incident needs to be associated with dispatch information and dispatch information cannot exist without an incident.

There is a one-to-one relationship between dispatch information and service area as the dispatch information table is only associated with one service area table. It is mandatory for both sides as dispatch information needs to be associated with service area and vice versa.

There is a one-to-one relationship between dispatch information and dispatch as dispatch information only requires one dispatch and vice versa. It is mandatory for both sides as dispatch information needs to be associated with dispatch and vice versa.

There is a one-to-one relationship between dispatch and ambulance as the dispatch only needs one ambulance. It is mandatory for dispatch to require an ambulance as it needs one to assist in patient care, but an ambulance does not have to be associated with a dispatch (optional).

There is a one-to-many relationship between dispatch and paramedic as the dispatch needs one or many paramedics to dispatch to the incident. It is mandatory for dispatch to require paramedics as it needs them to aid in dispatch, but a paramedic does not have to be associated with a dispatch (optional).

There is a one-to-one relationship between dispatch and cardiac treatment facility as dispatch is only associated with one instance of cardiac treatment facility being true or false. It is mandatory for both sides as dispatch needs to be associated with Cardiac Treatment Facility and vice versa.

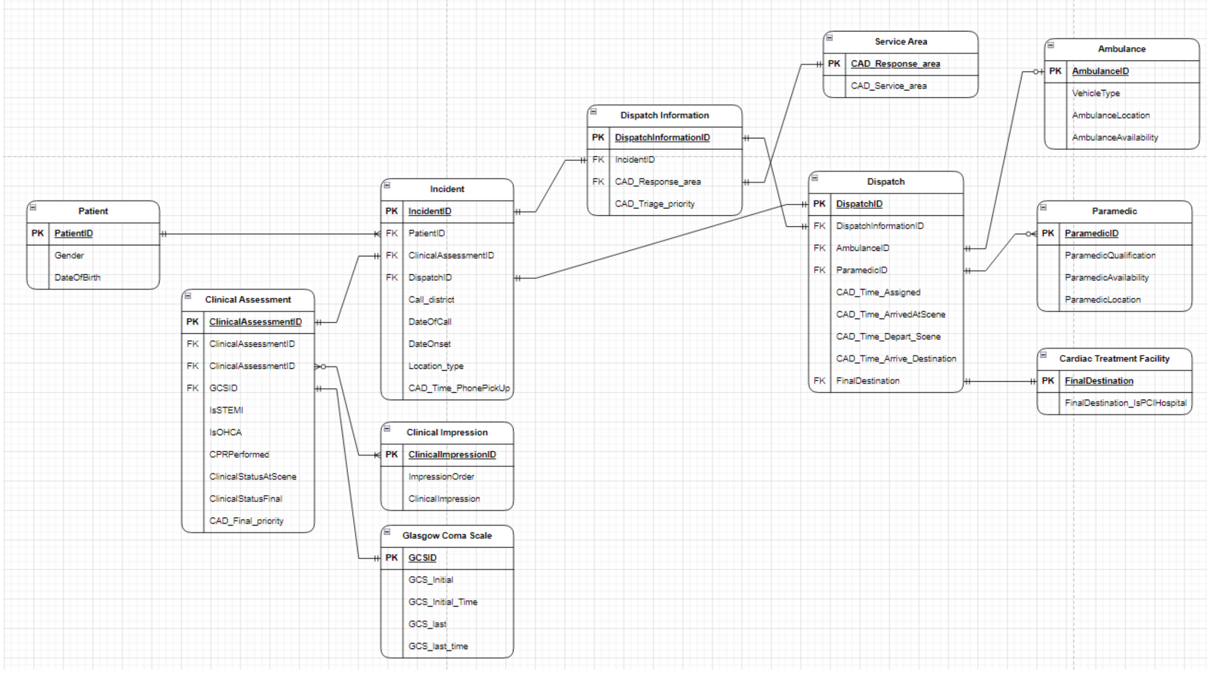
There is a one-to-one relationship between clinical assessment and incident as each incident can only be associated with one clinical assessment. It is mandatory for both sides as an incident needs to have a clinical assessment record and a clinical assessment cannot exist without a incident.

There is a one-to-many relationship between clinical assessment and clinical impression as each incident can be associated with many clinical assessments. It is mandatory for both sides as a clinical assessment needs clinical impressions, and a clinical impression can not exist without a clinical assessment.

There is a one-to-one relationship between clinical assessment and Glasgow coma scale as each clinical assessment can only be associated with one instance of Glasgow Coma Scale (being both the initial and final times). It is mandatory for both sides as a clinical assessment needs to have the GCS measures and times, and the Glasgow Coma Scale measures cannot exist without a clinical assessment.

There is a one-to-one relationship between incident and dispatch as each incident require a dispatch of ambulance and paramedics and there is only one dispatch per incident. It is mandatory for both sides as an incident needs dispatch and dispatch does not exist in the first place without an incident.

ER Diagram representing the database schema:



Database Summary:

The database meets the data requirements by capturing and storing a wide array of necessary information including patient details, incident details, important times, etc. The database is also free of data redundancy, is normalised, and has maintains data integrity.

The database also adequately meets the business needs of the EMS board.

Dispatching Efficiency:

Using the dispatch information, which consists of relevant attributes such as triage priority and the location of the incident, the appropriate dispatch can be allocated by allocating the most appropriate ambulance type (both in terms of the ambulance location and facilities) and paramedics to the incident based on the CAD triage priority. This way, the suitable ambulance with the correct paramedics can be dispatched to the incident promptly.

Dispatching Accuracy:

The designed database system prioritises incidents by their triage priority (the priority allocated to the incident from the phone call from the patient to the EMS) in the dispatch information table, which results in the dispatch table (which has relationships with the ambulance and paramedics tables) sending prompt and accurate services to the incident. This results in incidents with a higher triage priority receiving care first (and with the efficient dispatching as described in the dispatching efficiency business requirement).

Overall, the new and improved database now is efficient and effective and meets the business and data requirements of the EMS board and more. It provides an excellent foundation and should fulfil the further needs of the EMS board such as implementing the database schema into SQL to run queries on.

Submission Two

SQL Implementation

<u>Implementation of the database schema in a suitable relational database management system</u>

To successfully implement my database schema in a suitable relational database management system, I decided to MySQL.

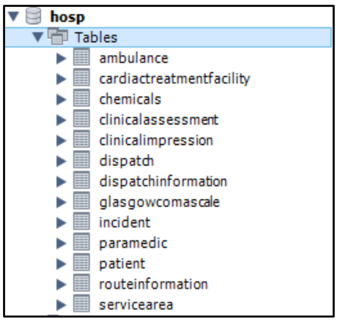
Some reasons why I used MySQL for my relational database management system:

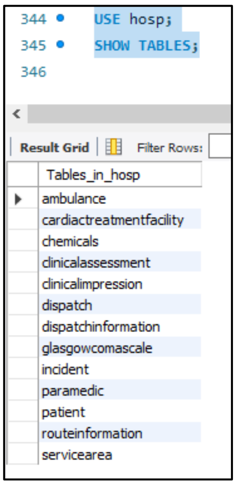
- **Reliability**
  Due to the high-stakes nature of the services that the EMS provides, it is of utmost importance that the RDMS system used be reliable. MySQL is a very reliable RDMS which is used by millions and has a great reputation in terms of reliability (https://www.mysql.com/industry/faq/).
- **Speed**
  MySQL is fast and efficient in processing data and retrieving records. This is very important for the EMS as they need to be able to quickly access and process data under time pressure in emergency scenarios.
- **Security**
  One of the main advantages of MySQL over other RDMSs is its security capabilities. Security is highly important for the EMS as they are dealing with private and sensitive medical information.
- **Cost**
  MySQL is free using the community edition. This makes it very suitable for the EMS as it means that they do not have to pay anything for their RDMS (MySQL). Because of this, they can focus more of their spending on the life-saving care that they provide. However, paying for MySQL (eg MySQL Enterprise) and other cloud services may be necessary if wanting more advanced capabilities. Even then, MySQL is quite cost-competitive.
- **Data Warehouse Support**
  MySQL offers data warehouse support. Having a data warehouse is very important for managing and analysing large amounts of data. This feature enables the storage, retrieval, and analysis of historical and real-time data, empowering the EMS team to make data-driven decisions, optimize dispatch processes, and improve emergency medical services.

To implement the database schema into MySQL, I used my schema from submission one (which was marked very well). I changed a few small things about the original schema for its implementation into SQL in an effort to comply better with normalisation as well as function better within MySQL. I also made some larger changes to the schema to fulfil the further requirements wanted by the EMS in the SQL implementation (determining routes, chemical information, etc.).

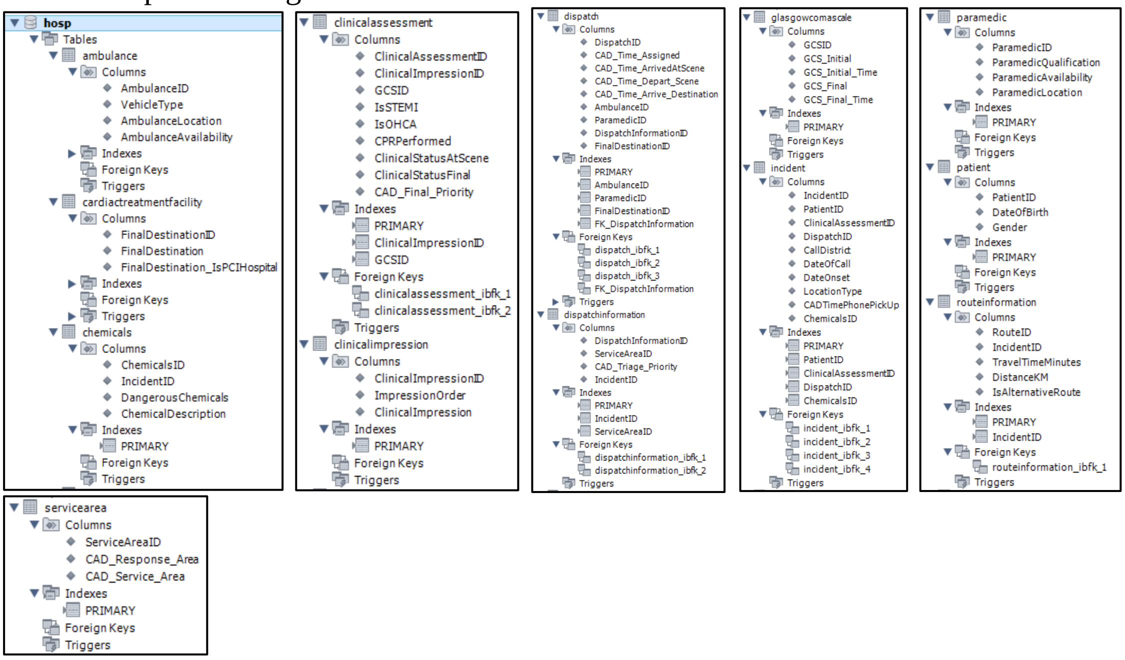
To show that I have successfully implemented the database schema in a suitable relational database management system, I will provide screenshots from my (successful) MySQL implementation.

General showing of tables:

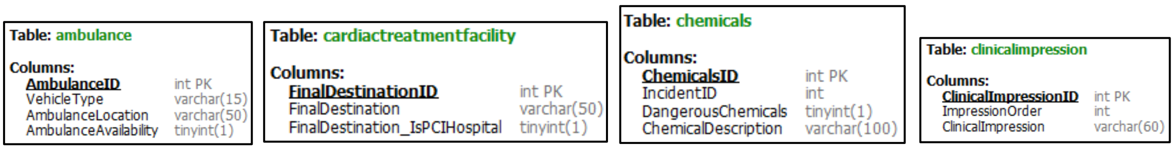


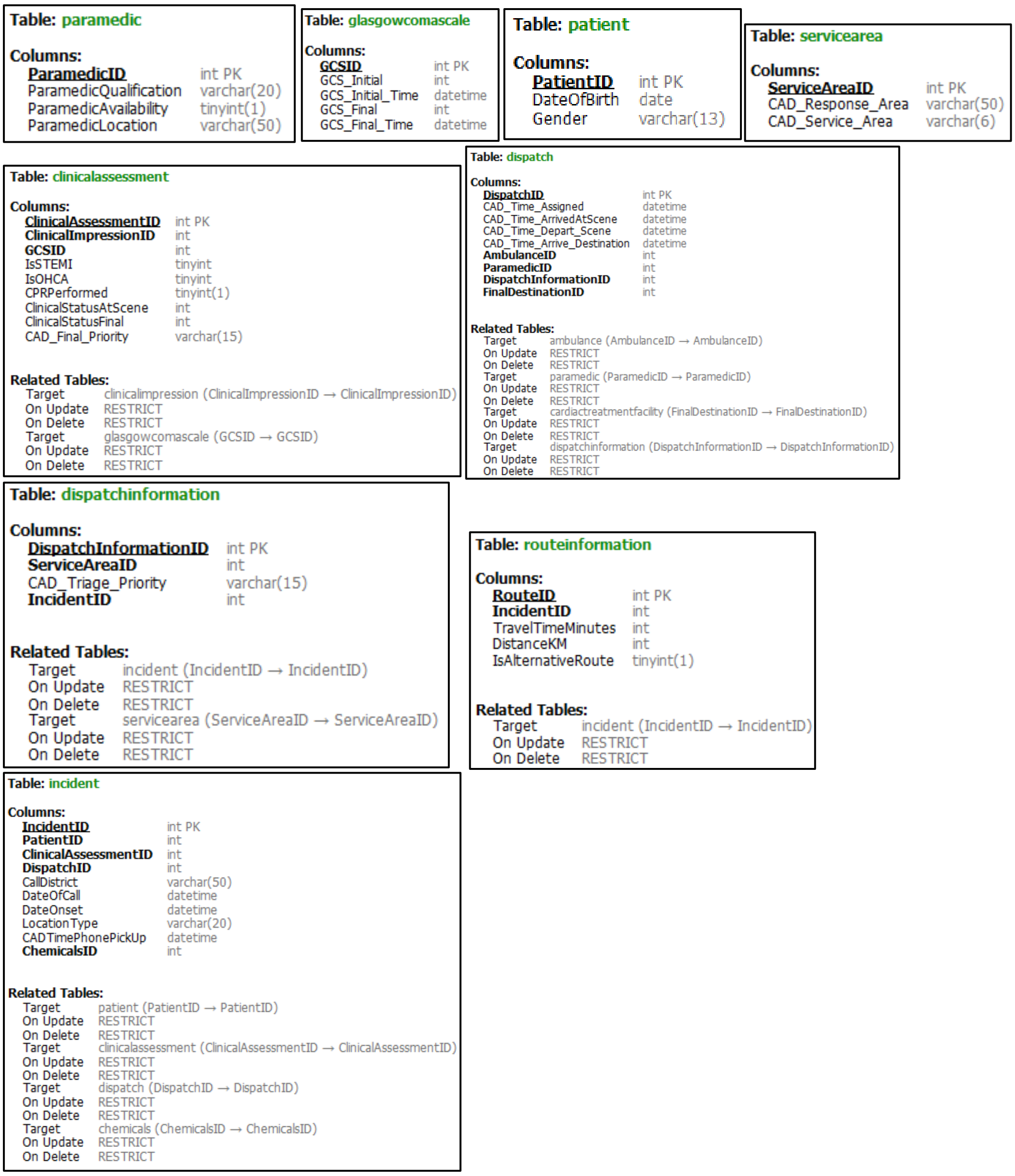


More in-depth showing of tables:



More table information:





Create the necessary tables and populate them with sample data

The SQL code used to implement the schema in MySQL

```sql
CREATE TABLE Patient (
    PatientID INT PRIMARY KEY,
    DateOfBirth DATE NOT NULL,
    Gender VARCHAR(13) CHECK(Gender IN ('Male', 'Female', 'Indeterminate')) NOT NULL
);

CREATE TABLE ClinicalImpression (
    ClinicalImpressionID INT PRIMARY KEY,
    ImpressionOrder INT NOT NULL,
    ClinicalImpression VARCHAR(60)
);

CREATE TABLE GlasgowComaScale (
    GCSID INT PRIMARY KEY,
```

```
    GCS_Initial INT,
    GCS_Initial_Time DATETIME NOT NULL,
    GCS_Final INT,
    GCS_Final_Time DATETIME
);

CREATE TABLE ClinicalAssessment (
    ClinicalAssessmentID INT PRIMARY KEY,
    ClinicalImpressionID INT NOT NULL,
    GCSID INT NOT NULL,
    IsSTEMI TINYINT CHECK(IsSTEMI IN (1, 0)) NOT NULL,
    IsOHCA TINYINT CHECK(IsOHCA IN (1, 0)) NOT NULL,
    CPRPerformed BOOLEAN NOT NULL,
    ClinicalStatusAtScene INT CHECK (ClinicalStatusAtScene >= 0 AND ClinicalStatusAtScene
<= 4),
    ClinicalStatusFinal INT CHECK (ClinicalStatusFinal >= 0 AND ClinicalStatusFinal <= 4),
    CAD_Final_Priority VARCHAR(15) CHECK (CAD_Final_Priority IN ('Purple', 'Red', 'Red 1',
'Red 2', 'Orange', 'Orange 1', 'Orange 2', 'Green', 'Green 1', 'Green 2', 'Grey', 'PTS',
'Private Hire', 'Air Transfer', 'Notification', 'Triage')) NOT NULL,
    FOREIGN KEY (ClinicalImpressionID) REFERENCES ClinicalImpression(ClinicalImpressionID),
    FOREIGN KEY (GCSID) REFERENCES GlasgowComaScale(GCSID)
);

CREATE TABLE Incident (
    IncidentID INT PRIMARY KEY,
    PatientID INT NOT NULL,
    ClinicalAssessmentID INT NOT NULL,
    DispatchID INT NOT NULL,
    CallDistrict VARCHAR(50) NOT NULL,
    DateOfCall DATETIME NOT NULL,
    DateOnset DATETIME,
    LocationType VARCHAR(20) NOT NULL,
    CADTimePhonePickUp DATETIME,
    ChemicalsID INT NOT NULL,
    FOREIGN KEY (PatientID) REFERENCES Patient(PatientID),
    FOREIGN KEY (ClinicalAssessmentID) REFERENCES ClinicalAssessment(ClinicalAssessmentID),
    FOREIGN KEY (DispatchID) REFERENCES Dispatch(DispatchID),
    FOREIGN KEY (ChemicalsID) REFERENCES Chemicals(ChemicalsID)
);

CREATE TABLE DispatchInformation (
    DispatchInformationID INT PRIMARY KEY,
    ServiceAreaID INT NOT NULL,
    CAD_Triage_Priority VARCHAR(15) CHECK (CAD_Triage_Priority IN ('Purple', 'Red', 'Red
1', 'Red 2', 'Orange', 'Orange 1', 'Orange 2', 'Green', 'Green 1', 'Green 2', 'Grey',
'PTS', 'Private Hire', 'Air Transfer', 'Notification', 'Triage')) NOT NULL,
    IncidentID INT NOT NULL,
    FOREIGN KEY (IncidentID) REFERENCES Incident(IncidentID),
    FOREIGN KEY (ServiceAreaID) REFERENCES ServiceArea(ServiceAreaID)
);

CREATE TABLE ServiceArea (
        ServiceAreaID INT PRIMARY KEY,
    CAD_Response_Area VARCHAR(50) NOT NULL,
    CAD_Service_Area VARCHAR(6) CHECK (CAD_Service_Area IN ('urban', 'rural', 'remote'))
NOT NULL
);
```

```sql
CREATE TABLE Ambulance (
    AmbulanceID INT PRIMARY KEY,
    VehicleType VARCHAR(15) CHECK (VehicleType IN ('Air Ambulance', 'CRT', 'DELTA', 'EAS',
'ECHO', 'EVENTS', 'FRU', 'HYBRID', 'MGR', 'PTS', 'ROMEO', 'SIERRA', 'TANGO', 'UCC')) NOT
NULL,
    AmbulanceLocation VARCHAR(50) NOT NULL,
    AmbulanceAvailability BOOLEAN NOT NULL
);

CREATE TABLE Paramedic (
    ParamedicID INT PRIMARY KEY,
    ParamedicQualification VARCHAR(20) CHECK (ParamedicQualification IN ('ECP', 'ICP',
'Other ATP/No ATP', 'Para', 'EMT')) NOT NULL,
    ParamedicAvailability BOOLEAN NOT NULL,
    ParamedicLocation VARCHAR(50) NOT NULL
);

CREATE TABLE CardiacTreatmentFacility (
          FinalDestinationID INT PRIMARY KEY NOT NULL,
    FinalDestination VARCHAR(50),
    FinalDestination_IsPCIHospital BOOLEAN NOT NULL
);

CREATE TABLE Dispatch (
    DispatchID INT PRIMARY KEY,
    CAD_Time_Assigned DATETIME,
    CAD_Time_ArrivedAtScene DATETIME,
    CAD_Time_Depart_Scene DATETIME,
    CAD_Time_Arrive_Destination DATETIME,
    AmbulanceID INT NOT NULL,
    ParamedicID INT NOT NULL,
    DispatchInformationID INT NOT NULL,
    FinalDestinationID INT NOT NULL,
    FOREIGN KEY (AmbulanceID) REFERENCES Ambulance(AmbulanceID),
    FOREIGN KEY (ParamedicID) REFERENCES Paramedic(ParamedicID),
    FOREIGN KEY (FinalDestinationID) REFERENCES
CardiacTreatmentFacility(FinalDestinationID)
);

ALTER TABLE Dispatch
ADD CONSTRAINT FK_DispatchInformation
FOREIGN KEY (DispatchInformationID) REFERENCES DispatchInformation(DispatchInformationID);

CREATE TABLE RouteInformation (
    RouteID INT PRIMARY KEY,
    IncidentID INT,
    TravelTimeMinutes INT,
    DistanceKM INT,
    IsAlternativeRoute BOOLEAN NOT NULL,
    FOREIGN KEY (IncidentID) REFERENCES Incident(IncidentID)
);

CREATE TABLE Chemicals (
  ChemicalsID INT PRIMARY KEY,
  IncidentID INT NOT NULL,
  DangerousChemicals BOOLEAN,
```

```sql
  ChemicalDescription VARCHAR(100)
);
```

Sample data for the tables. For sample demonstration purposes, I kept the number of sample rows low as well as kept the data values quite simple as not to overcomplicate things. In the real-world context of the EMS, there will be *many* more rows for each table.

```sql
INSERT INTO Chemicals (ChemicalsID, IncidentID, DangerousChemicals, ChemicalDescription)
VALUES
    (1, 1, 1, 'Chemical spill in a healthcare facility'),
    (2, 2, 0, 'N/A'),
    (3, 3, 1, 'Toxic fumes at a home'),
    (4, 4, 1, 'Hazardous materials storage facility at an aged care facility'),
    (5, 5, 0, 'N/A'),
    (6, 6, 0, 'N/A'),
    (7, 7, 1, 'Chemical exposure at a home'),
    (8, 8, 1, 'Chemical spill in a healthcare facility'),
    (9, 9, 0, 'N/A'),
    (10, 10, 1, 'Chemical leak');

INSERT INTO RouteInformation (RouteID, IncidentID, TravelTimeMinutes, DistanceKM,
IsAlternativeRoute)
VALUES
    (1, 1, 20, 24, 0),
    (2, 1, 30, 41, 1),
    (3, 2, 15, 11, 0),
    (4, 2, 20, 15, 1),
    (5, 3, 25, 36, 0),
    (6, 3, 40, 56, 1),
    (7, 4, 10, 9, 0),
    (8, 4, 15, 13, 1),
    (9, 5, 30, 46, 0),
    (10, 5, 35, 52, 1),
    (11, 6, 25, 32, 0),
    (12, 6, 50, 73, 1),
    (13, 7, 15, 22, 0),
    (14, 7, 30, 45, 1),
    (15, 8, 10, 12, 0),
    (16, 8, 20, 24, 1),
    (17, 9, 25, 35, 0),
    (18, 9, 35, 50, 1),
    (19, 10, 40, 61, 0),
    (20, 10, 55, 85, 1);

INSERT INTO Patient (PatientID, DateOfBirth, Gender)
VALUES
    (1, '1990-05-15', 'Male'),
    (2, '1985-08-21', 'Female'),
    (3, '2000-03-10', 'Indeterminate'),
    (4, '1978-12-03', 'Male'),
    (5, '1995-04-28', 'Female'),
    (6, '1980-09-17', 'Indeterminate'),
    (7, '2002-11-25', 'Male'),
    (8, '1993-07-14', 'Female'),
    (9, '1987-02-09', 'Male'),
```

```
    (10, '1999-10-06', 'Female');

INSERT INTO ClinicalImpression (ClinicalImpressionID, ImpressionOrder, ClinicalImpression)
VALUES
    (1, 1, 'Cardiac chest pain'),
    (2, 1, 'Renal Colic'),
    (3, 1, 'Cardiac chest pain'),
    (4, 1, 'Palliative care'),
    (5, 1, 'Anaphylaxis'),
    (6, 1, 'Acute low back pain'),
    (7, 1, 'Intentional Poisoning'),
    (8, 1, 'Fall minor injury'),
    (9, 1, 'Cardiac chest pain'),
    (10, 1, 'Atypical chest pain');

INSERT INTO GlasgowComaScale (GCSID, GCS_Initial, GCS_Initial_Time, GCS_Final,
GCS_Final_Time)
VALUES
    (1, 10, '2023-10-10 08:00:00', 13, '2023-10-10 08:30:00'),
    (2, 12, '2023-10-10 09:15:00', 15, '2023-10-10 10:00:00'),
    (3, 8, '2023-10-10 10:30:00', 10, '2023-10-10 11:00:00'),
    (4, 9, '2023-10-10 11:45:00', 14, '2023-10-10 12:15:00'),
    (5, 14, '2023-10-10 13:30:00', 15, '2023-10-10 14:00:00'),
    (6, 7, '2023-10-10 14:30:00', 12, '2023-10-10 15:00:00'),
    (7, 13, '2023-10-10 15:45:00', 15, '2023-10-10 16:30:00'),
    (8, 11, '2023-10-10 17:15:00', 14, '2023-10-10 17:45:00'),
    (9, 15, '2023-10-10 18:30:00', 15, '2023-10-10 19:00:00'),
    (10, 6, '2023-10-10 19:30:00', 8, '2023-10-10 20:00:00');

INSERT INTO ClinicalAssessment (ClinicalAssessmentID, ClinicalImpressionID, GCSID, IsSTEMI,
IsOHCA, CPRPerformed, ClinicalStatusAtScene, ClinicalStatusFinal, CAD_Final_Priority)
VALUES
    (1, 1, 1, 1, 0, 1, 3, 2, 'Purple'),
    (2, 2, 2, 0, 1, 0, 1, 3, 'Red'),
    (3, 3, 3, 0, 0, 0, 2, 2, 'Orange 1'),
    (4, 4, 4, 1, 0, 1, 1, 4, 'Green'),
    (5, 5, 5, 1, 1, 1, 0, 0, 'Triage'),
    (6, 6, 6, 0, 0, 0, 4, 1, 'PTS'),
    (7, 7, 7, 1, 0, 1, 2, 0, 'Red 2'),
    (8, 8, 8, 0, 1, 0, 3, 3, 'Grey'),
    (9, 9, 9, 1, 0, 1, 1, 2, 'Orange'),
    (10, 10, 10, 0, 1, 0, 2, 4, 'Private Hire');

INSERT INTO CardiacTreatmentFacility (FinalDestinationID, FinalDestination,
FinalDestination_IsPCIHospital)
VALUES
    (1, 'Hawkes Bay Hospital', 1),
    (2, 'Dunstan Hospital', 0),
    (3, 'Fitzroy Medical Centre', 0),
    (4, 'NA', 0),
    (5, 'Kew Hospital (Invercargill Hosp)', 0),
    (6, 'Bay of Islands Hospital', 0),
    (7, 'Dunedin Public Hospital', 0),
    (8, 'Northland Base Hospital', 0),
    (9, 'Middlemore Hospital', 1),
    (10, 'NA', 0);
```

```sql
INSERT INTO Ambulance (AmbulanceID, VehicleType, AmbulanceLocation, AmbulanceAvailability)
VALUES
    (1, 'Air Ambulance', 'Location 1', 1),
    (2, 'ECHO', 'Location 2', 0),
    (3, 'PTS', 'Location 3', 1),
    (4, 'DELTA', 'Location 4', 0),
    (5, 'UCC', 'Location 5', 1),
    (6, 'TANGO', 'Location 6', 0),
    (7, 'MGR', 'Location 7', 1),
    (8, 'CRT', 'Location 8', 0),
    (9, 'EVENTS', 'Location 9', 1),
    (10, 'FRU', 'Location 10', 0);

INSERT INTO Paramedic (ParamedicID, ParamedicQualification, ParamedicAvailability,
ParamedicLocation)
VALUES
    (1, 'ECP', 1, 'Location 1'),
    (2, 'ICP', 0, 'Location 2'),
    (3, 'Para', 1, 'Location 3'),
    (4, 'EMT', 1, 'Location 4'),
    (5, 'ECP', 0, 'Location 5'),
    (6, 'Other ATP/No ATP', 1, 'Location 6'),
    (7, 'Para', 0, 'Location 7'),
    (8, 'ICP', 1, 'Location 8'),
    (9, 'EMT', 0, 'Location 9'),
    (10, 'Other ATP/No ATP', 1, 'Location 10');

INSERT INTO DispatchInformation (DispatchInformationID, ServiceAreaID, CAD_Triage_Priority,
IncidentID)
VALUES
    (1, 1, 'Red', 1),
    (2, 2, 'Orange', 2),
    (3, 3, 'Purple', 3),
    (4, 4, 'Green', 4),
    (5, 5, 'Red 2', 5),
    (6, 6, 'PTS', 6),
    (7, 7, 'Orange 1', 7),
    (8, 8, 'Grey', 8),
    (9, 9, 'Red 1', 9),
    (10, 10, 'Triage', 10);

INSERT INTO ServiceArea (ServiceAreaID, CAD_Response_Area, CAD_Service_Area)
VALUES
    (1, 'Hastings', 'Urban'),
    (2, 'Cromwell', 'Rural'),
    (3, 'Great Barrier Island', 'Remote'),
    (4, 'Palmerston North', 'Urban'),
    (5, 'Winton', 'Rural'),
    (6, 'Kaikohe', 'Remote'),
    (7, 'Dunedin', 'Urban'),
    (8, 'Whangarei', 'Urban'),
    (9, 'Botany', 'Urban'),
    (10, 'Blenheim', 'Urban');

INSERT INTO Dispatch (DispatchID, CAD_Time_Assigned, CAD_Time_ArrivedAtScene,
CAD_Time_Depart_Scene, CAD_Time_Arrive_Destination, FinalDestinationID, AmbulanceID,
ParamedicID, DispatchInformationID)
```

```sql
VALUES
    (1, '2023-10-10 08:00:00', '2023-10-10 08:20:00', '2023-10-10 08:40:00', '2023-10-10 09:00:00', 1, 1, 1, 1),
    (2, '2023-10-10 09:30:00', '2023-10-10 09:45:00', '2023-10-10 10:00:00', '2023-10-10 10:15:00', 2, 2, 2, 2),
    (3, '2023-10-10 10:45:00', '2023-10-10 11:00:00', '2023-10-10 11:15:00', '2023-10-10 11:30:00', 3, 3, 3, 3),
    (4, '2023-10-10 12:00:00', '2023-10-10 12:15:00', '2023-10-10 12:30:00', '2023-10-10 12:45:00', 4, 4, 4, 4),
    (5, '2023-10-10 13:30:00', '2023-10-10 13:45:00', '2023-10-10 14:00:00', '2023-10-10 14:15:00', 5, 5, 5, 5),
    (6, '2023-10-10 14:45:00', '2023-10-10 15:00:00', '2023-10-10 15:15:00', '2023-10-10 15:30:00', 6, 6, 6, 6),
    (7, '2023-10-10 16:00:00', '2023-10-10 16:15:00', '2023-10-10 16:30:00', '2023-10-10 16:45:00', 7, 7, 7, 7),
    (8, '2023-10-10 17:15:00', '2023-10-10 17:30:00', '2023-10-10 17:45:00', '2023-10-10 18:00:00', 8, 8, 8, 8),
    (9, '2023-10-10 18:30:00', '2023-10-10 18:45:00', '2023-10-10 19:00:00', '2023-10-10 19:15:00', 9, 9, 9, 9),
    (10, '2023-10-10 19:45:00', '2023-10-10 20:00:00', '2023-10-10 20:15:00', '2023-10-10 20:30:00', 10, 10, 10, 10);

INSERT INTO Incident (IncidentID, PatientID, ClinicalAssessmentID, DispatchID, ChemicalsID, CallDistrict, DateOfCall, DateOnset, LocationType, CADTimePhonePickUp)
VALUES
    (1, 1, 1, 1, 1, 'Central South', '2023-10-10 08:00:00', '2023-10-10 08:15:00', 'Healthcare Facility', '2023-10-10 08:10:00'),
    (2, 2, 2, 2, 2, 'Southland/Otago', '2023-10-10 09:30:00', '2023-10-10 09:45:00', 'Home', '2023-10-10 09:35:00'),
    (3, 3, 3, 3, 3, 'Auckland', '2023-10-10 10:45:00', '2023-10-10 11:00:00', 'Home', '2023-10-10 10:50:00'),
    (4, 4, 4, 4, 4, 'Central South', '2023-10-10 12:00:00', '2023-10-10 12:15:00', 'Aged Care Facility', '2023-10-10 12:05:00'),
    (5, 5, 5, 5, 5, 'Southland/Otago', '2023-10-10 13:30:00', '2023-10-10 13:45:00', 'Road', '2023-10-10 13:35:00'),
    (6, 6, 6, 6, 6, 'Northland', '2023-10-10 14:45:00', '2023-10-10 15:00:00', 'Footpath', '2023-10-10 14:50:00'),
    (7, 7, 7, 7, 7, 'Southland/Otago', '2023-10-10 16:00:00', '2023-10-10 16:15:00', 'Home', '2023-10-10 16:05:00'),
    (8, 8, 8, 8, 8, 'Northland', '2023-10-10 17:15:00', '2023-10-10 17:30:00', 'Healthcare Facility', '2023-10-10 17:20:00'),
    (9, 9, 9, 9, 9, 'Auckland', '2023-10-10 18:30:00', '2023-10-10 18:45:00', 'Workplace', '2023-10-10 18:35:00'),
    (10, 10, 10, 10, 10, 'Tasman', '2023-10-10 19:45:00', '2023-10-10 20:00:00', 'Aged Care Facility', '2023-10-10 19:50:00');
```

Populated tables in MySQL:

```
342 ● SELECT * FROM Patient;
```

PatientID	DateOfBirth	Gender
1	1990-05-15	Male
2	1985-08-21	Female
3	2000-03-10	Indeterminate
4	1978-12-03	Male
5	1995-04-28	Female
6	1980-09-17	Indeterminate
7	2002-11-25	Male
8	1993-07-14	Female
9	1987-02-09	Male
10	1999-10-06	Female

```
343 ● SELECT * FROM ClinicalImpression;
344 ● SELECT * FROM GlasgowComaScale;
```

ClinicalImpressionID	ImpressionOrder	ClinicalImpression
1	1	Cardiac chest pain
2	1	Renal Colic
3	1	Cardiac chest pain
4	1	Palliative care
5	1	Anaphylaxis
6	1	Acute low back pain
7	1	Intentional Poisoning
8	1	Fall minor injury
9	1	Cardiac chest pain
10	1	Atypical chest pain
NULL	NULL	NULL

```
344 ● SELECT * FROM GlasgowComaScale;
345 ● SELECT * FROM ClinicalAssessment;
```

GCSID	GCS_Initial	GCS_Initial_Time	GCS_Final	GCS_Final_Time
1	10	2023-10-10 08:00:00	13	2023-10-10 08:30:00
2	12	2023-10-10 09:15:00	15	2023-10-10 10:00:00
3	8	2023-10-10 10:30:00	10	2023-10-10 11:00:00
4	9	2023-10-10 11:45:00	14	2023-10-10 12:15:00
5	14	2023-10-10 13:30:00	15	2023-10-10 14:00:00
6	7	2023-10-10 14:30:00	12	2023-10-10 15:00:00
7	13	2023-10-10 15:45:00	15	2023-10-10 16:30:00
8	11	2023-10-10 17:15:00	14	2023-10-10 17:45:00
9	15	2023-10-10 18:30:00	15	2023-10-10 19:00:00
10	6	2023-10-10 19:30:00	8	2023-10-10 20:00:00
NULL	NULL	NULL	NULL	NULL

```
345 ● SELECT * FROM ClinicalAssessment;
346 ● SELECT * FROM Incident;
```

ClinicalAssessmentID	ClinicalImpressionID	GCSID	IsSTEMI	IsOHCA	CPRPerformed	ClinicalStatusAtScene	ClinicalStatusFinal	CAD_Final_Priority
1	1	1	1	0	1	3	2	Purple
2	2	2	0	1	0	1	3	Red
3	3	3	0	0	0	2	2	Orange 1
4	4	4	1	0	1	1	4	Green
5	5	5	1	1	1	0	0	Triage
6	6	6	0	0	0	4	1	PTS
7	7	7	1	0	1	2	0	Red 2
8	8	8	0	1	0	3	3	Grey
9	9	9	1	0	1	1	2	Orange
10	10	10	0	1	0	2	4	Private Hire
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
348 ● SELECT * FROM ServiceArea;
349 ● SELECT * FROM Ambulance;
```

ServiceAreaID	CAD_Response_Area	CAD_Service_Area
1	Hastings	Urban
2	Cromwell	Rural
3	Great Barrier Island	Remote
4	Palmerston North	Urban
5	Winton	Rural
6	Kaikohe	Remote
7	Dunedin	Urban
8	Whangarei	Urban
9	Botany	Urban
10	Blenheim	Urban

```
346 ● SELECT * FROM Incident;
347 ● SELECT * FROM DispatchInformation;
```

IncidentID	PatientID	ClinicalAssessmentID	DispatchID	CallDistrict	DateOfCall	DateOnset	LocationType	CADTimePhonePickUp	ChemicalsID
1	1	1	1	Central South	2023-10-10 08:00:00	2023-10-10 08:15:00	Healthcare Facility	2023-10-10 08:10:00	1
2	2	2	2	Southland/Otago	2023-10-10 09:30:00	2023-10-10 09:45:00	Home	2023-10-10 09:35:00	2
3	3	3	3	Auckland	2023-10-10 10:45:00	2023-10-10 11:00:00	Home	2023-10-10 10:50:00	3
4	4	4	4	Central South	2023-10-10 12:00:00	2023-10-10 12:15:00	Aged Care Facility	2023-10-10 12:05:00	4
5	5	5	5	Southland/Otago	2023-10-10 13:30:00	2023-10-10 13:45:00	Road	2023-10-10 13:35:00	5
6	6	6	6	Northland	2023-10-10 14:45:00	2023-10-10 15:00:00	Footpath	2023-10-10 14:50:00	6
7	7	7	7	Southland/Otago	2023-10-10 16:00:00	2023-10-10 16:15:00	Home	2023-10-10 16:05:00	7
8	8	8	8	Northland	2023-10-10 17:15:00	2023-10-10 17:30:00	Healthcare Facility	2023-10-10 17:20:00	8
9	9	9	9	Auckland	2023-10-10 18:30:00	2023-10-10 18:45:00	Workplace	2023-10-10 18:35:00	9
10	10	10	10	Tasman	2023-10-10 19:45:00	2023-10-10 20:00:00	Aged Care Facility	2023-10-10 19:50:00	10

```
349 ● SELECT * FROM Ambulance;
350 ● SELECT * FROM Paramedic;
```

AmbulanceID	VehicleType	AmbulanceLocation	AmbulanceAvailability
1	Air Ambulance	Location 1	1
2	ECHO	Location 2	0
3	PTS	Location 3	1
4	DELTA	Location 4	0
5	UCC	Location 5	1
6	TANGO	Location 6	0
7	MGR	Location 7	1
8	CRT	Location 8	0
9	EVENTS	Location 9	1
10	FRU	Location 10	0

```
350 ● SELECT * FROM Paramedic;
351 ● SELECT * FROM CardiacTreatmentFacility;
```

ParamedicID	ParamedicQualification	ParamedicAvailability	ParamedicLocation
1	ECP	1	Location 1
2	ICP	0	Location 2
3	Para	1	Location 3
4	EMT	1	Location 4
5	ECP	0	Location 5
6	Other ATP/No ATP	1	Location 6
7	Para	0	Location 7
8	ICP	1	Location 8
9	EMT	0	Location 9
10	Other ATP/No ATP	1	Location 10

```
351 ● SELECT * FROM CardiacTreatmentFacility;
352 ● SELECT * FROM Dispatch;
```

FinalDestinationID	FinalDestination	FinalDestination_IsPCIHospital
1	Hawkes Bay Hospital	1
2	Dunstan Hospital	0
3	Fitzroy Medical Centre	0
4	NA	0
5	Kew Hospital (Invercargill Hosp)	0
6	Bay of Islands Hospital	0
7	Dunedin Public Hospital	0
8	Northland Base Hospital	0
9	Middlemore Hospital	1
10	NA	0

```
352 ● SELECT * FROM Dispatch;
353 ● SELECT * FROM RouteInformation;
```

DispatchID	CAD_Time_Assigned	CAD_Time_ArrivedAtScene	CAD_Time_Depart_Scene	CAD_Time_Arrive_Destination	AmbulanceID	ParamedicID	DispatchInformationID	FinalDestinationID
1	2023-10-10 08:00:00	2023-10-10 08:20:00	2023-10-10 08:40:00	2023-10-10 09:00:00	1	1	1	1
2	2023-10-10 09:30:00	2023-10-10 09:45:00	2023-10-10 10:00:00	2023-10-10 10:15:00	2	2	2	2
3	2023-10-10 10:45:00	2023-10-10 11:00:00	2023-10-10 11:15:00	2023-10-10 11:30:00	3	3	3	3
4	2023-10-10 12:00:00	2023-10-10 12:15:00	2023-10-10 12:30:00	2023-10-10 12:45:00	4	4	4	4
5	2023-10-10 13:30:00	2023-10-10 13:45:00	2023-10-10 14:00:00	2023-10-10 14:15:00	5	5	5	5
6	2023-10-10 14:45:00	2023-10-10 15:00:00	2023-10-10 15:15:00	2023-10-10 15:30:00	6	6	6	6
7	2023-10-10 16:00:00	2023-10-10 16:15:00	2023-10-10 16:30:00	2023-10-10 16:45:00	7	7	7	7
8	2023-10-10 17:15:00	2023-10-10 17:30:00	2023-10-10 17:45:00	2023-10-10 18:00:00	8	8	8	8
9	2023-10-10 18:30:00	2023-10-10 18:45:00	2023-10-10 19:00:00	2023-10-10 19:15:00	9	9	9	9
10	2023-10-10 19:45:00	2023-10-10 20:00:00	2023-10-10 20:15:00	2023-10-10 20:30:00	10	10	10	10

```
353 ● SELECT * FROM RouteInformation;
354 ● SELECT * FROM Chemicals;
```

RouteID	IncidentID	TravelTimeMinutes	DistanceKM	IsAlternativeRoute
1	1	20	24	0
2	1	30	41	1
3	2	15	11	0
4	2	20	15	1
5	3	25	36	0
6	3	40	56	1
7	4	10	9	0
8	4	15	13	1
9	5	30	46	0
10	5	35	52	1
11	6	25	32	0
12	6	50	73	1
13	7	15	22	0
14	7	30	45	1
15	8	10	12	0
16	8	20	24	1
17	9	25	35	0
18	9	35	50	1
19	10	40	61	0
20	10	55	85	1

```
360 ● SELECT * FROM DispatchInformation;
```

DispatchInformationID	ServiceAreaID	CAD_Triage_Priority	IncidentID
1	1	Red	1
2	2	Orange	2
3	3	Purple	3
4	4	Green	4
5	5	Red 2	5
6	6	PTS	6
7	7	Orange 1	7
8	8	Grey	8
9	9	Red 1	9
10	10	Triage	10
NULL	NULL	NULL	NULL

```
354 ● SELECT * FROM Chemicals;
```

ChemicalsID	IncidentID	DangerousChemicals	ChemicalDescription
1	1	1	Chemical spill in a healthcare facility
2	2	0	N/A
3	3	1	Toxic fumes at a home
4	4	1	Hazardous materials storage facility at an aged ...
5	5	0	N/A
6	6	0	N/A
7	7	1	Chemical exposure at a home
8	8	1	Chemical spill in a healthcare facility
9	9	0	N/A
10	10	1	Chemical leak

Note: I later added a ‘route name’ attribute to the route information table and added sample data accordingly.

```
ALTER TABLE RouteInformation
ADD RouteName VARCHAR(10);

UPDATE RouteInformation
```

```sql
SET RouteName = CASE
    WHEN RouteID = 1 THEN 'Route A'
    WHEN RouteID = 2 THEN 'Route B'
    WHEN RouteID = 3 THEN 'Route C'
    WHEN RouteID = 4 THEN 'Route D'
    WHEN RouteID = 5 THEN 'Route E'
    WHEN RouteID = 6 THEN 'Route F'
    WHEN RouteID = 7 THEN 'Route G'
    WHEN RouteID = 8 THEN 'Route H'
    WHEN RouteID = 9 THEN 'Route I'
    WHEN RouteID = 10 THEN 'Route J'
    WHEN RouteID = 11 THEN 'Route K'
    WHEN RouteID = 12 THEN 'Route L'
    WHEN RouteID = 13 THEN 'Route M'
    WHEN RouteID = 14 THEN 'Route N'
    WHEN RouteID = 15 THEN 'Route O'
    WHEN RouteID = 16 THEN 'Route P'
    WHEN RouteID = 17 THEN 'Route Q'
    WHEN RouteID = 18 THEN 'Route R'
    WHEN RouteID = 19 THEN 'Route S'
    WHEN RouteID = 20 THEN 'Route T'
END;
```

```sql
SELECT *
FROM routeinformation;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

RouteID	IncidentID	TravelTimeMinutes	DistanceKM	IsAlternativeRoute	RouteName
1	1	20	24	0	Route A
2	1	30	41	1	Route B
3	2	15	11	0	Route C
4	2	20	15	1	Route D
5	3	25	36	0	Route E
6	3	40	56	1	Route F
7	4	10	9	0	Route G
8	4	15	13	1	Route H
9	5	30	46	0	Route I
10	5	35	52	1	Route J
11	6	25	32	0	Route K
12	6	50	73	1	Route L
13	7	15	22	0	Route M
14	7	30	45	1	Route N
15	8	10	12	0	Route O
16	8	20	24	1	Route P
17	9	25	35	0	Route Q
18	9	35	50	1	Route R
19	10	40	61	0	Route S
20	10	55	85	1	Route T

Write SQL queries to demonstrate the following functionalities:

I will now show my query code and its output to demonstrate the following functionalities:

a) Retrieving incident locations and details
b) Determining the quickest and alternative routes to an incident
c) Retrieving information about premises and dangerous chemicals
d) Generating detailed statistics on response times and callout causes

Retrieving incident locations and details:

To do this, I used a simple 'SELECT * FROM' operation on the incident table to show all the attributes from the incident table. This includes showing both the incident location (in the form of the call district and location type) as well as the other incident details (date of call, date onset, etc.).

```
SELECT *
FROM INCIDENT;
```

```
SELECT *
FROM INCIDENT;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

IncidentID	PatientID	ClinicalAssessmentID	DispatchID	CallDistrict	DateOfCall	DateOnset	LocationType	CADTimePhonePickUp	ChemicalsID
1	1	1	1	Central South	2023-10-10 08:00:00	2023-10-10 08:15:00	Healthcare Facility	2023-10-10 08:10:00	1
2	2	2	2	Southland/Otago	2023-10-10 09:30:00	2023-10-10 09:45:00	Home	2023-10-10 09:35:00	2
3	3	3	3	Auckland	2023-10-10 10:45:00	2023-10-10 11:00:00	Home	2023-10-10 10:50:00	3
4	4	4	4	Central South	2023-10-10 12:00:00	2023-10-10 12:15:00	Aged Care Facility	2023-10-10 12:05:00	4
5	5	5	5	Southland/Otago	2023-10-10 13:30:00	2023-10-10 13:45:00	Road	2023-10-10 13:35:00	5
6	6	6	6	Northland	2023-10-10 14:45:00	2023-10-10 15:00:00	Footpath	2023-10-10 14:50:00	6
7	7	7	7	Southland/Otago	2023-10-10 16:00:00	2023-10-10 16:15:00	Home	2023-10-10 16:05:00	7
8	8	8	8	Northland	2023-10-10 17:15:00	2023-10-10 17:30:00	Healthcare Facility	2023-10-10 17:20:00	8
9	9	9	9	Auckland	2023-10-10 18:30:00	2023-10-10 18:45:00	Workplace	2023-10-10 18:35:00	9
10	10	10	10	Tasman	2023-10-10 19:45:00	2023-10-10 20:00:00	Aged Care Facility	2023-10-10 19:50:00	10

Additionally, if let's say the EMS needed to know the details of incident 7, they could use the query:

```
SELECT *
FROM INCIDENT
WHERE incidentid = 7;
```

```
SELECT *
FROM INCIDENT
WHERE incidentid = 7;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

IncidentID	PatientID	ClinicalAssessmentID	DispatchID	CallDistrict	DateOfCall	DateOnset	LocationType	CADTimePhonePickUp	ChemicalsID
7	7	7	7	Southland/Otago	2023-10-10 16:00:00	2023-10-10 16:15:00	Home	2023-10-10 16:05:00	7
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Or if the EMS wanted to know the incident location and details where the call district location is Northland they could use:

```
SELECT *
FROM INCIDENT
WHERE calldistrict = 'Northland';
```

```
SELECT *
FROM INCIDENT
WHERE calldistrict = 'Northland';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

IncidentID	PatientID	ClinicalAssessmentID	DispatchID	CallDistrict	DateOfCall	DateOnset	LocationType	CADTimePhonePickUp	ChemicalsID
6	6	6	6	Northland	2023-10-10 14:45:00	2023-10-10 15:00:00	Footpath	2023-10-10 14:50:00	6
8	8	8	8	Northland	2023-10-10 17:15:00	2023-10-10 17:30:00	Healthcare Facility	2023-10-10 17:20:00	8

<u>Determining the quickest and alternative routes to an incident:</u>

To do this, a simple 'SELECT * FROM' statement can be used on the routeinformation table. By doing this, it shows both the routes, – those being the the non-alternative route (the quickest rouse) and the alternative rouse – for each incident. Additionally, it shows the distance in km for each route as well as the time that route would take.

```
SELECT *
FROM routeinformation JOIN incident on routeinformation.incidentID =
incident.incidentID;
```

```sql
SELECT *
FROM routeinformation JOIN incident on routeinformation.incidentID = incident.incidentID;
```

RouteID	IncidentID	TravelTimeMinutes	DistanceKM	IsAlternativeRoute	RouteName	IncidentID	PatientID	ClinicalAssessmentID	DispatchID	CallDistrict	DateOfCall	DateOnset	LocationType	CADTimePhonePickUp	Chemicals
1	1	20	24	0	Route A	1	1	1	1	Central South	2023-10-10 08:00:00	2023-10-10 08:15:00	Healthcare Facility	2023-10-10 08:10:00	1
2	1	30	41	1	Route B	1	1	1	1	Central South	2023-10-10 08:00:00	2023-10-10 08:15:00	Healthcare Facility	2023-10-10 08:10:00	1
3	2	15	11	0	Route C	2	2	2	2	Southland/Otago	2023-10-10 09:30:00	2023-10-10 09:45:00	Home	2023-10-10 09:35:00	2
4	2	20	15	1	Route D	2	2	2	2	Southland/Otago	2023-10-10 09:30:00	2023-10-10 09:45:00	Home	2023-10-10 09:35:00	2
5	3	25	36	0	Route E	3	3	3	3	Auckland	2023-10-10 10:45:00	2023-10-10 11:00:00	Home	2023-10-10 10:50:00	3
6	3	40	56	1	Route F	3	3	3	3	Auckland	2023-10-10 10:45:00	2023-10-10 11:00:00	Home	2023-10-10 10:50:00	3
7	4	10	9	0	Route G	4	4	4	4	Central South	2023-10-10 12:00:00	2023-10-10 12:15:00	Aged Care Facility	2023-10-10 12:05:00	4
8	4	15	13	1	Route H	4	4	4	4	Central South	2023-10-10 12:00:00	2023-10-10 12:15:00	Aged Care Facility	2023-10-10 12:05:00	4
9	5	30	46	0	Route I	5	5	5	5	Southland/Otago	2023-10-10 13:30:00	2023-10-10 13:45:00	Road	2023-10-10 13:35:00	5
10	5	35	52	1	Route J	5	5	5	5	Southland/Otago	2023-10-10 13:30:00	2023-10-10 13:45:00	Road	2023-10-10 13:35:00	5
11	6	25	32	0	Route K	6	6	6	6	Northland	2023-10-10 14:45:00	2023-10-10 15:00:00	Footpath	2023-10-10 14:50:00	6
12	6	50	73	1	Route L	6	6	6	6	Northland	2023-10-10 14:45:00	2023-10-10 15:00:00	Footpath	2023-10-10 14:50:00	6
13	7	15	22	0	Route M	7	7	7	7	Southland/Otago	2023-10-10 16:00:00	2023-10-10 16:15:00	Home	2023-10-10 16:05:00	7
14	7	30	45	1	Route N	7	7	7	7	Southland/Otago	2023-10-10 16:00:00	2023-10-10 16:15:00	Home	2023-10-10 16:05:00	7
15	8	10	12	0	Route O	8	8	8	8	Northland	2023-10-10 17:15:00	2023-10-10 17:30:00	Healthcare Facility	2023-10-10 17:20:00	8
16	8	20	24	1	Route P	8	8	8	8	Northland	2023-10-10 17:15:00	2023-10-10 17:30:00	Healthcare Facility	2023-10-10 17:20:00	8
17	9	25	35	0	Route Q	9	9	9	9	Auckland	2023-10-10 18:30:00	2023-10-10 18:45:00	Workplace	2023-10-10 18:35:00	9
18	9	35	50	1	Route R	9	9	9	9	Auckland	2023-10-10 18:30:00	2023-10-10 18:45:00	Workplace	2023-10-10 18:35:00	9
19	10	40	61	0	Route S	10	10	10	10	Tasman	2023-10-10 19:45:00	2023-10-10 20:00:00	Aged Care Facility	2023-10-10 19:50:00	10
20	10	55	85	1	Route T	10	10	10	10	Tasman	2023-10-10 19:45:00	2023-10-10 20:00:00	Aged Care Facility	2023-10-10 19:50:00	10

Or if they don't need the all the incident information with the routes, they can simply use:

```sql
SELECT *
FROM routeinformation;
```

```sql
SELECT *
FROM routeinformation;
```

RouteID	IncidentID	TravelTimeMinutes	DistanceKM	IsAlternativeRoute	RouteName
1	1	20	24	0	Route A
2	1	30	41	1	Route B
3	2	15	11	0	Route C
4	2	20	15	1	Route D
5	3	25	36	0	Route E
6	3	40	56	1	Route F
7	4	10	9	0	Route G
8	4	15	13	1	Route H
9	5	30	46	0	Route I
10	5	35	52	1	Route J
11	6	25	32	0	Route K
12	6	50	73	1	Route L
13	7	15	22	0	Route M
14	7	30	45	1	Route N
15	8	10	12	0	Route O
16	8	20	24	1	Route P
17	9	25	35	0	Route Q
18	9	35	50	1	Route R
19	10	40	61	0	Route S
20	10	55	85	1	Route T

In a practical context, let's say that the EMS needed the alternative route for incident 3 because there was a road closure on the quickest route. In this case, they could use the query:

```sql
SELECT routename, distancekm, TravelTimeMinutes
FROM routeinformation JOIN incident on routeinformation.incidentID = incident.incidentID
WHERE incident.incidentid = 3 and isalternativeroute is TRUE;
```

This results in the EMS receiving the route name (so that they know what route to take), as well as additional details such as distance and travel time.

```sql
SELECT routename, distancekm, TravelTimeMinutes
FROM routeinformation
WHERE incidentid = 3 and isalternativeroute is TRUE;
```

routename	distancekm	TravelTimeMinutes
Route F	56	40

<u>Retrieving information about premises and dangerous chemicals:</u>

To do this, I used a simple 'SELECT * FROM' operation on the chemicals table joined with the incident table to retrieve whether or not an incident (premise) has dangerous chemicals involved or not and if so, a short description of the chemical. It also gives other incident (premise information)

```
SELECT *
FROM chemicals JOIN incident on chemicals.incidentID =
incident.incidentID;
```

```
SELECT *
FROM chemicals JOIN incident on chemicals.incidentID = incident.incidentID;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ChemicalsID	IncidentID	DangerousChemicals	ChemicalDescription	IncidentID	PatientID	ClinicalAssessmentID	DispatchID	CallDistrict	DateOfCall	DateOnset	LocationType	CADTimePhonePickUp	ChemicalsID
1	1	1	Chemical spill in a healthcare facility	1	1	1	1	Central South	2023-10-10 08:00:00	2023-10-10 08:15:00	Healthcare Facility	2023-10-10 08:10:00	1
2	2	0	N/A	2	2	2	2	Southland/Otago	2023-10-10 09:30:00	2023-10-10 09:45:00	Home	2023-10-10 09:35:00	2
3	3	1	Toxic fumes at a home	3	3	3	3	Auckland	2023-10-10 10:45:00	2023-10-10 11:00:00	Home	2023-10-10 10:50:00	3
4	4	1	Hazardous materials storage facility at an aged ...	4	4	4	4	Central South	2023-10-10 12:00:00	2023-10-10 12:15:00	Aged Care Facility	2023-10-10 12:05:00	4
5	5	0	N/A	5	5	5	5	Southland/Otago	2023-10-10 13:30:00	2023-10-10 13:45:00	Road	2023-10-10 13:35:00	5
6	6	0	N/A	6	6	6	6	Northland	2023-10-10 14:45:00	2023-10-10 15:00:00	Footpath	2023-10-10 14:50:00	6
7	7	1	Chemical exposure at a home	7	7	7	7	Southland/Otago	2023-10-10 16:00:00	2023-10-10 16:15:00	Home	2023-10-10 16:05:00	7
8	8	1	Chemical spill in a healthcare facility	8	8	8	8	Northland	2023-10-10 17:15:00	2023-10-10 17:30:00	Healthcare Facility	2023-10-10 17:20:00	8
9	9	0	N/A	9	9	9	9	Auckland	2023-10-10 18:30:00	2023-10-10 18:45:00	Workplace	2023-10-10 18:35:00	9
10	10	1	Chemical leak	10	10	10	10	Tasman	2023-10-10 19:45:00	2023-10-10 20:00:00	Aged Care Facility	2023-10-10 19:50:00	10

If the EMS wanted to look at all incidents involving dangerous chemicals, they could use the query:

```
SELECT *
FROM chemicals JOIN incident on chemicals.incidentID =
incident.incidentID
WHERE dangerouschemicals is TRUE;
```

```
SELECT *
FROM chemicals JOIN incident on chemicals.incidentID = incident.incidentID
WHERE dangerouschemicals is TRUE;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ChemicalsID	IncidentID	DangerousChemicals	ChemicalDescription	IncidentID	PatientID	ClinicalAssessmentID	DispatchID	CallDistrict	DateOfCall	DateOnset	LocationType	CADTimePhonePickUp	ChemicalsID
1	1	1	Chemical spill in a healthcare facility	1	1	1	1	Central South	2023-10-10 08:00:00	2023-10-10 08:15:00	Healthcare Facility	2023-10-10 08:10:00	1
3	3	1	Toxic fumes at a home	3	3	3	3	Auckland	2023-10-10 10:45:00	2023-10-10 11:00:00	Home	2023-10-10 10:50:00	3
4	4	1	Hazardous materials storage facility at an aged ...	4	4	4	4	Central South	2023-10-10 12:00:00	2023-10-10 12:15:00	Aged Care Facility	2023-10-10 12:05:00	4
7	7	1	Chemical exposure at a home	7	7	7	7	Southland/Otago	2023-10-10 16:00:00	2023-10-10 16:15:00	Home	2023-10-10 16:05:00	7
8	8	1	Chemical spill in a healthcare facility	8	8	8	8	Northland	2023-10-10 17:15:00	2023-10-10 17:30:00	Healthcare Facility	2023-10-10 17:20:00	8
10	10	1	Chemical leak	10	10	10	10	Tasman	2023-10-10 19:45:00	2023-10-10 20:00:00	Aged Care Facility	2023-10-10 19:50:00	10

Or if the EMS wanted to know the incidentids of all incidents where the chemical description is 'Chemical spill in a healthcare facility':

```
SELECT incidentid
FROM chemicals
WHERE ChemicalDescription = 'Chemical spill in a healthcare facility';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

incidentid
1
8

<u>Generating detailed statistics on response times and callout causes:</u>

To do this, I will give a few example queries.

Response times:

To get the average response time from date of call (by the patient) to the time arrived at the scene by the EMS:

```
SELECT AVG(TIMESTAMPDIFF(MINUTE, dateofcall, CAD_Time_ArrivedAtScene)) AS AVGCallToArrivalResponseTime
FROM Dispatch
JOIN Incident ON Dispatch.DispatchID = Incident.DispatchID;
```

```sql
SELECT AVG(TIMESTAMPDIFF(MINUTE, dateofcall, CAD_Time_ArrivedAtScene)) AS AVGCallToArrivalResponseTime
FROM Dispatch
JOIN Incident ON Dispatch.DispatchID = Incident.DispatchID;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

AVGCallToArrivalResponseTime
15.5000

To get the average response time from time assigned to time arrived at the scene (in this case it's the same as the previous result due to sample data, but again, this is just a demonstration):

```sql
SELECT AVG(TIMESTAMPDIFF(MINUTE, CAD_Time_Assigned, CAD_Time_ArrivedAtScene)) AS AVGAssignedToArrivalResponseTime
FROM Dispatch;
```

```sql
SELECT AVG(TIMESTAMPDIFF(MINUTE, CAD_Time_Assigned, CAD_Time_ArrivedAtScene)) AS AVGAssignedToArrivalResponseTime
FROM Dispatch;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

AVGAssignedToArrivalResponseTime
15.5000

We can also look at the average response time from date of call (by the patient) to the time arrived at the scene by the EMS where the location type is 'home':

```sql
SELECT AVG(TIMESTAMPDIFF(MINUTE, dateofcall, CAD_Time_ArrivedAtScene)) AS AVGHomeCallToArrivalResponseTime
FROM Dispatch
JOIN Incident ON Dispatch.DispatchID = Incident.DispatchID
WHERE locationtype = 'Home';
```

```sql
SELECT AVG(TIMESTAMPDIFF(MINUTE, dateofcall, CAD_Time_ArrivedAtScene)) AS AVGHomeCallToArrivalResponseTime
FROM Dispatch
JOIN Incident ON Dispatch.DispatchID = Incident.DispatchID
WHERE locationtype = 'Home';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

AVGHomeCallToArrivalResponseTime
15.0000

Callout causes:

Number of each CAD_triage_priority (resulting priority given from the callout call between patient and the EMS):

```sql
SELECT CAD_Triage_Priority, COUNT(*)
FROM incident JOIN dispatchinformation on incident.incidentid = dispatchinformation.incidentid
GROUP BY CAD_Triage_Priority;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

CAD_Triage_Priority	COUNT(*)
Red	1
Orange	1
Purple	1
Green	1
Red 2	1
PTS	1
Orange 1	1
Grey	1
Red 1	1
Triage	1

Number of each clinical impression type. Clinical impression are recorded by medical professionals when the EMS arrives, but the patient likely mentions such impressions as their callout cause as well:

```
SELECT clinicalimpression, COUNT(*)
FROM clinicalimpression
GROUP BY clinicalimpression;
```

Result Grid | Filter Rows: | Expor

clinicalimpression	COUNT(*)
Cardiac chest pain	3
Renal Colic	1
Palliative care	1
Anaphylaxis	1
Acute low back pain	1
Intentional Poisoning	1
Fall minor injury	1
Atypical chest pain	1

Data Quality

Identify potential data quality issues that can arise in the Ambulance and Paramedic Dispatch System.

I will first begin my discussion on data quality for the EMS by identifying and discussing some potential data quality issues that can arise in the Ambulance and Paramedic Dispatch System (APDS).

Accuracy: Accuracy is *highly* important for the EMS as inaccurate data can result in some very undesirable outcomes for them. The data quality issue of accuracy relates to the degree to which the stored data in the systems reflects the real/intended data. Inaccurate data can potentially arise in the APDS because of such factors as data entry errors by responders (eg hearing the wrong information through the emergency phone call) or data being outdated and thus inaccurate. A potential data quality accuracy issue in the APDS could be inaccurate location information data resulting in the EMS sending out an emergency vehicle to the wrong location and/or resulting in delays. Another example could be inaccurate priority level data, in which based on the stored (but inaccurate data), the EMS sends an under-equipped vehicle because they thought the priority level was understated (so not accurate). I could go on with many such examples regarding all the other attributes of the EMS systems here as well as the potential for inaccurate data is high. This data quality issue regarding accuracy also appears to not only be a potential issue, but a *current* issue for the APDS. This is because of their business need for 'dispatching accuracy' and their explanation that "…the dispatch process sometimes results in delays in responding to critical emergencies. The lack of advanced dispatch capabilities can lead to suboptimal resource allocation, impacting the delivery of life-saving care.". These issues presented here by the EMS seem to be highly related to the data quality issue of accuracy, hence it appears to be a current issue of theirs.

Volatility: Volatility in data quality refers to the degree to which data quality changes over time. This can potentially arise in the APDS as there will likely be a lot of data that may change over time and diminish in quality due to the fast-paced and rapidly changing nature of the EMS, as well as the large amount of data that they are dealing with in providing their services.

Completeness: Completeness is another potential data quality issue that can arise in the APDS and is related somewhat closely to the aforementioned issue of accuracy. One aspect of completeness is having no missing values for necessary and critical fields. This could arise in a number of ways in the APDS via human error by simply forgetting to enter a value, issues integrating/migrating data, and data corruption. As a potential example, there could be incidents in which critical information such as the location of the incident is missing. Similarly to the data quality issue of accuracy, critical data this would likely result in delays and confusion. Another aspect of completeness is that the necessary data is in the system to meet the demands of EMS. This can arise as an issue perhaps if the EMS department and those behind the EMS APDS system did not think to include a specific attribute. For example, the EMS may need to know something like what medication a patient was taking, but as this data was not stored in the system, it is a completeness issue and thus can result in inefficiencies in the EMS providing their care.

Timeliness: Timeliness is certainly another potential data quality issue that can arise in the APDS. Timeliness refers to the data being sufficiently up-to-date and accessible/available when it is needed. Some reasons for this issue potentially arising in the APDS could be that those working at the EMS are not updating the data fast enough (delays) or there are technical/system issues in which the data is not able to be transferred. A potential example of this could be that someone at the EMS forgets to update the data, and thus an ambulance is sent to an old location. Similarly to accuracy, this also seems to be a *current* data quality issue as the EMS department states in their case that they identify a business need in 'dispatching efficiency' and that response times need to be reduced. A large reason for these delays in dispatching may be that data is not being updated fast enough and when needed, hence, there are delays.

Consistency: Consistency in data quality refers to the degree to which data is reliable, uniform, and equal within and across datasets. Data quality issues surrounding consistency may arise in the APDS due to the many tables and attributes in the system. Like many other data quality issues, this can arise due to human data entry mistakes as well as the dataset system being poorly set up.

<u>Propose strategies to ensure data quality, including validation, cleansing, and governance practices.</u>

I will now propose some actionable strategies which the EMS can use to ensure data quality, including validation, cleansing, and governance practices.

Data validation:
One strategy to ensure data quality and validate the data is implementing data validation rules into the APDS. Data validation rules are conditions that the data must meet in order for the data to be considered valid data (hence the term 'data validation'). These rules help ensure the accuracy and integrity in of data. Data validation rules can include checking that the inputted data is the correct data type/format, that the data is not null (missing), and that data values are within a specified range. These validation rules can be automated in the APDS, making it a very simple yet effective strategy for ensuring data quality. Validation practices

such as data validation rules can help combat some of the potential/current data quality issues previously mentioned such as accuracy and completeness.

Data cleansing:
Data cleansing is the process of identifying and fixing/removing corrupt or incorrect records from a dataset and is arguably the most critical aspect of ensuring data quality. To implement this strategy, the EMS should first identify the data fields which are critical for the EMS to provide their service. This would likely include many fields such as the locations of the incidents, clinical impressions, etc. as these are all critical fields for the EMS's operations. Then the EMS should monitor this data by recording where the errors are coming from and analyse trends relating to this. By doing this, the EMS can identify the errors efficiently and these errors can later be cleansed. Now that the errors have been identified, the EMS should begin resolving inaccuracies by identifying and deleting duplicate values. Then missing values should be resolved by filling them in so that there are no 'gaps' in the dataset as these can cause significant issues if not resolved. It is also important that the EMS standardises its data cleansing process as this will make this whole cleansing process much more efficient. This is very important for the EMS due to the fast-paced, high-stakes nature of their services. This means formulating/creating a data cleansing process that is easily replicable by determining such factors as which data is used most frequently, how often data will be cleansed, etc. This process should be shared widely within the EMS team to keep everyone informed and on the same page. Finally, this data cleansing process should be improved and adapted over time to continue to meet the EMS' needs sufficiently.

Data integration:
Poor integration of data from different sources can result in low data quality. The EMS team should first come up with a comprehensive data integration plan to mitigate such issues. This plan should outline the EMS' goals, objectives, and desired outcomes as well as identify the sources of the data. Then the EMS should profile the data which needs to be integrated in an effort to understand if there are any issues in the data and what its structure and overall quality are. Then data mapping should be done to determine how the data from each source will be transformed and how it will be entered into the desired data store. Finally, with much oversight, all the planning done here should implemented, hopefully resulting in an integrated data store which is void of data quality issues. Note: This was a very simplified explanation of data integration, more will be covered in a later section.

Establishing a data governance framework:
The EMS should establish a well-structured data governance committee/council that is in charge of defining data quality standards and enforcing these standards (more on this in the next points).

Data quality standards:
Once the EMS has established such a framework, they should define and establish clear data quality standards that define aspects of the expected/desired level of data quality within the EMS. Such standards can help ensure accuracy, reliability, consistency, and more in the data. By explicitly defining such policies and standards, the EMS can ensure over time that their data meets these standards and adjust accordingly.

Data stewardship:
The EMS should choose some people to be data stewards who are responsible for ensuring data quality for different aspects/sets of the EMS's data. This means ensuring data quality

standards are met as defined by the EMS. By doing so, the defined data quality standards can be enforced.

Create a data dictionary and other data documentation:
Some of the people on the data governance team should work to create a data dictionary which contains data definitions and metadata. Additionally, thorough documentation of other aspects of data quality also be done. These documents and the data dictionary should be shared widely amongst the EMS team so that everyone is on the same page. By doing so, data quality can be ensured as it means that everyone in the team can reference the documents when making decisions related to the data.

Data quality awareness amongst the EMS:
Those at the higher levels of the EMS should try to stress the importance of data quality to those working at all levels of the organisation. By doing so, the workers can be more aware of the importance of good data quality, and thus, will be more likely to ensure that good data quality is being maintained.

Discuss the importance of data quality in maintaining an effective dispatch system.

Ensuring a high standard of data quality is important in all organisations that deal with data. However, in the context of the EMS and their dispatch system, this importance in data quality is of even greater significance. This is because the EMS dispatch system is dealing with high-stakes, fast-paced emergency situations where patients' well-being and lives are at stake. I will now explain some reasons why data quality is important in maintaining an effective dispatch system.

Response times:
The EMS stated in the case that they are struggling with response times due to delays. This may be caused at least in part by a lack of data quality. Good data quality is *critical* in delivering fast response times as it means that the EMS can respond to emergencies effectively without having to worry about issues relating to the data such as its accuracy or reliability. If the data quality is lacking, this could result in delays due to the EMS having to operate off of inaccurate/missing/outdated data. Response times are undoubtedly one of the most important aspects of an effective dispatch system, and without good data quality, it would be very difficult to be effective in this regard.

Patient and Public Satisfaction:
One of the core measures of a dispatch system being effective is if the individual patients and the public as a whole are satisfied with the service. Accurate, reliable, and complete data is essential to deliver the best possible service to each patient, thus, improving the overall public trust in the service. A lack of data quality could result in outcomes which lead to poor patient experiences and erode public trust.

Resource allocation:
The EMS also stated in the case that they are struggling with suboptimal resource allocation. Again, this may be due to poor data quality. Resource allocation is highly important for the EMS and maintaining an effective dispatch system. Firstly, the EMS has limited resources (paramedics, vehicles, etc.) so they need to be allocated effectively (they can't just send all the highest-level ambulances and paramedics to every incident because they only have so

many). Secondly, correct resource allocation is essential as the wrong resources for an incident can negatively impact the care delivered (for example, sending an underequipped ambulance). If the data is inaccurate/missing/unreliable, the EMS' resources may be allocated incorrectly, resulting in less optimal service by the EMS. By ensuring high-quality data, the EMS can make sure that resources are allocated well without having to deal with the allocation problems that arise with low-quality data.

Legal/regulatory requirements:
As the EMS is dealing with sensitive data, they are almost certainly subject to relevant laws and regulations. Ensuring that the dispatch system data is of high quality (accurate/not missing, etc.) is very important for meeting legal and regulatory requirements. It is possible that if a dispatch system has very poor data quality, it may be subject to legal investigations and penalties. This may erode public trust and ultimately make the dispatch system as effective as it can be.

Better outcomes amongst the team behind the dispatch system:
While many of the measures of an effective dispatch system look at the outcomes of the patient, it is also important to consider those who work as part of the dispatch team themselves. The impacts of poor data quality in a dispatch system could lead to a lot of frustration and anger amongst workers, ultimately resulting in a lower standard of care being delivered. By ensuring good data quality, the dispatch system team can worry less about data issues and focus more on the task at hand – saving lives. Hence, the dispatch system would be more effective.

Overall safety and patient outcomes:
At the end of the day, the most important measure of a dispatch system being effective is that such a system is improving safety and patient outcomes (saving lives) in the relevant communities it operates in. By ensuring good data quality, this is possible, and it reduces the chance of delays and data-related mistakes from happening.

Utilisation of Data

Discuss how the data collected by the Ambulance and Paramedic Dispatch System can be combined with big data to enhance emergency response and prevention.

By combining the data collected by the APDS with big data, the EMS' emergency response and prevention can be greatly enhanced. Doing so can address some of the business needs and issues that the EMS currently faces and help them provide a better quality of service overall.

Real-time tracking/monitoring:
Combining the data collected by the APDS with big data can enable real-time monitoring of ambulances and paramedics. This would allow others on the EMS team to track details in real-time such as the location and availability of the ambulances and paramedics. This would help ensure that the closest, most equipped ambulance and paramedics are sent to each incident and may greatly help with speeding up response times.

Trends over time:

Using big data with the collected data by the APDS can uncover insightful trends and patterns in the APDS data over time. The insights gained from these trends can be used to make informed, data-driven decisions by the EMS team that could be used to enhance their emergency response and the prevention of emergencies.

Predicting outcomes:
Predictive analytics involves using past and present data to predict future trends and events. Predictive analytics (enabled by big data) could be used to help predict where emergencies will happen, what patient outcomes will be, or even the level of traffic (if the EMS were to collect traffic data). This would help greatly in emergency response as, based on predictions, the EMS could station ambulances/paramedics in areas where there are predicted to be emergencies, thus improving response times. Predictive analytics would also help greatly in prevention as the EMS could view the predictions generated and adjust their services appropriately so that they are better equipped for the future.

Preventative alerts:
A preventative alert system could be developed to enhance prevention measures for the EMS when there is a very concerning trend that the EMS needs to know about as soon as possible. The system would regularly analyse the data to check if there are any worrying trends such as a recent spike in patient symptoms being all the same type (like all the same disease). This could then alert the EMS before some situation (like a disease outbreak) gets too out of hand and unmanageable. Hence, it would enhance prevention.

Public health management:
Big data could be used for public health management as a preventative measure for the EMS. This could be done by analysing large amounts of relevant population data collected by the APDS in order to recognise any important details or trends across the population. For example, if there appeared growing trend that poor dietary choices were being made widely amongst the public, the EMS could run a public health campaign in an effort to prevent more of this from happening.

Evaluating and monitoring the EMS' performance:
Big data analytics could be applied to the APDS data to evaluate and monitor the performance of the EMS (in terms of response times, patient outcomes, etc.) over time. If the EMS were to make positive changes in response to this evaluation and monitoring, this would undoubtedly enhance their emergency response.

<u>Provide examples of big data technologies and techniques that can be applied to the Ambulance and Paramedic Dispatch System.</u>

NoSQL databases:
NoSQL databases are very suited for big data (as opposed to relational databases). NoSQL databases can be used for storing and managing large volumes of unstructured data and they are highly flexible and scalable. NoSQL databases can be applied to the APDS as a way to store and manage the EMS' big data. Specific NoSQL database technologies include MongoDB, Redis, and Cassandra.

Machine learning:
Machine learning algorithms can be used by the EMS for predictive analytics. By applying machine learning techniques to the APDS, future outcomes can be intelligently predicted.

Specific machine learning technologies include Apache Spark, Hadoop (with machine learning libraries and tools), and TensorFlow.

Geospatial Analytics:
As the APDS is dealing with a great amount of location data, it may make sense to apply the big data technology of geospatial analytics to the APDS. By doing so, location data can be effectively analysed and visualised. For example, visualising areas where there are a large amount of emergencies. Specific big data geospatial analytics tools include ArcGIS, PostGIS, and CARTO.

MapReduce:
MapReduce is a programming model used for large datasets and big data. It is highly scalable, flexible, simple, and has great performance. MapReduce could be applied to the APDS for processes such as data cleaning and integration of big data. MapReduce is not the best option as a real-time system, so it would not be suitable for some of the more immediate big data aspects of the APDS.

Data Visualisation:
Big data visualisation tools can be applied to the APDS to provide real-time information and insights presented in a visually appealing way. Data visualisations can help greatly in better representing the data so that the EMS team can more effectively act on the insights such visualisations bring. Specific data visualisation technologies/tools include Power BI, Tableau, and Datawrapper.

IoT (Internet of Things):
IoT integration with the APDS could provide a great amount of real-time data for the EMS. This integration could mean putting IoT devices in the ambulances, at locations, and with the paramedics themselves. By doing so, this big data that is generated could be analysed and tracked, leading to an enhancement in the EMS service. IoT is very broad and is made up of many components, but some IoT technologies include SAP Internet of Things, Microsoft Azure IoT Central, and Google Cloud IoT Core.

<u>Explain the potential benefits and challenges of integrating big data with the existing system.</u>

Potential benefits of integrating big data with the existing system:

Note: I covered quite a few benefits in the previous points, so I will be reiterating some of them here in this section.

- **Reducing costs**
  One potential benefit of integrating big data with the existing system is reducing costs. These cost savings could come in many such ways (due to big data integration) such as:
    - Saving money on fuel using route optimisation
    - Predicting future outcomes and the EMS team making budget decisions based on this information

  - Efficient allocation of resources. This could mean ambulances being closer to emergency hotspots to save on fuel or allocating paramedics effectively so that fewer paramedics are required.

- **Faster response times**
  Faster response times are a potential benefit of integrating big data with the existing system. Faster response times could result because of:
  - Route optimisation -- using big data to determine the best routes (given the circumstances) for the vehicles to travel to and from incidents.
  - Better allocation of resources such as paramedics and ambulances to be closer to where emergencies take place.
  - Traffic monitoring to choose the best route (could be considered part of route optimisation)

- **Better real-time data capabilities**
  Due to the fast-paced, real-time, and high-stakes nature of the EMS' services, real-time information and data processing is key to successfully running their operations. Integrating big data with the current system can potentially help improve the EMS' real-time data capabilities, possibly providing better real-time data on:
  - Location data such as real-time locations of the EMS vehicles and the incidents
  - On-the-fly route optimisation: routes can be updated in real-time to be as optimal as possible. As an example, there could be a car crash on the determined route of a vehicle to an incident. Using real-time route optimisation, this may be able to be quickly picked up by the system and result in the route being changed in rapid time.

- **Improved resource allocation**
  Big data technologies can be used to potentially better allocate the EMS' limited resources. Resource allocation was listed as a big issue for EMS in the given case, so improving this could be very useful for the EMS. Examples could include:
  - Analysing peak/busy times and days so that more paramedics, vehicles, and other resources can be utilised on those days.
  - Analysing the paramedic staff to utilise them more effectively for each incident.
  - Better tracking of resources. Big data can be used to track resources more effectively, thus allowing the EMS team to make more informed decisions relating to resource allocation.

- **Predicting future outcomes**
  One major potential benefit of integrating big data with the existing system could be predictive analytics. I will reiterate my previous point on predictive analytics here:

  Predictive analytics (enabled by big data) could be used to help predict where emergencies will happen, what patient outcomes will be, or even the level of traffic (if the EMS were to collect traffic data). This would help greatly in emergency response as, based off predictions, the EMS could station ambulances/paramedics in areas where there are predicted to be emergencies, thus improving response times. Predictive analytics would also help greatly in prevention as the EMS could view the predictions generated and adjust their services appropriately so that they are better equipped for the future.

- **Better patient outcomes**
  Because of some of the other benefits listed above, as well as some of the other points mentioned in this section on ‘utilisation of data’, the EMS may potentially be able to provide a better service than they currently do, and thus, patient outcomes will improve.

Potential challenges of integrating big data with the existing system:

- **Cost**
  While *reducing* costs is certainly a potential benefit, it also may be the opposite that costs increase. This is because there will likely be costs in paying for things such as hardware/software/cloud services, training or hiring staff, ongoing maintenance costs, and more.

- **Data quality**
  Due to the sheer amount of data in big data, it can be difficult to maintain high data quality compared to standard databases with smaller volumes of data. As previously stated, poor data quality can result in negative outcomes such as delays for the EMS.

- **The integration process**
  The integration process of integrating big data with the existing system could take a long time and there would likely be many hiccups along the way. It would likely require a lot of data cleaning/processing/wrangling which can be very time-consuming.

- **Issues with scalability**
  Integrating big data with the existing system will likely result in a very large amount of data being generated. If big data is not integrated well enough, it may not be able to scale well with the amount of data generated and cause issues.

- **Security**
  Data security is very important for the EMS as they are dealing with sensitive medical records. In integrating big data with the existing system, they would have to be very precise with the security measures as even the smallest vulnerabilities can result in data breaches.

- **Training**
  As briefly touched on in the ‘cost’ challenge, by integrating big data with the existing system, the EMS would likely have to retrain all personnel who work with the existing system to be able to effectively use the big data integrated system.

- **Staff unhappiness**
  Going off the last point, some of the EMS staff may be unhappy with the new big data system due to having to learn new things and possibly not being able to interact with it as well as they could with the previous system.

Data Environment and Management Plan

Discuss the concept of a data warehouse and its relevance to the Ambulance and Paramedic Dispatch System.

A data warehouse is a large central repository of data. It can be used to store, manage, and analyse data from different sources in an organisation. Data warehouses play a key role in business intelligence (BI) and can provide useful insights for an organisation, allowing an organisation to make more effective and data-driven decisions. I will now discuss some aspects of data warehouses in more detail:

Data integration:
Data warehouses collect many different types of data from various sources within an organisation. This process of integrating the data into a data warehouse is known as the Extract, Transform, Load (ETL) process. I will cover the ETL process in more detail soon.

Non-volatile:
Data warehouses are non-volatile. This means that once data is successfully in a data warehouse, it usually is not modified in any way. This helps ensure that historical data is accurate, accessible, and unaltered. By doing so, better and more accurate historical analysis and reporting can be done. Additionally, not altering stored data helps with data compliance, as altering past data can defy legal and regulatory requirements.

Subject-oriented:
Data warehouse data is organised and oriented by subject – usually, each subject being a particular area that the organisation deems meaningful. For example, suppliers, customers, sales, etc. Data warehouses being subject-oriented makes them more suited for modelling and analysis of data, ultimately making it good for data decision-making.

Time-variant:
Data warehouses are time-variant. This means that they record and maintain historical data and changes over time. This is as opposed to some other types of databases in which only current data may be stored. Data warehouses being time-variant makes them suitable for looking at trends over time and making data-driven decisions based on this historical data.

How data warehouses are relevant to the APDS:

- **Data integration and centralisation**
  The APDS likely has *many* different data sources consisting of a large number of data types and formats. This could include hospital data, phone data, location data, etc. Integrating and centralising this data into a data warehouse would help greatly with the overall data management and analysis.
- **Security**
  As I have mentioned many times in this report, security is very important for the EMS and the APDS system as they are dealing with sensitive medical information. A data warehouse which is designed with a high level of security can be an effective way to maintain data security. This is because storing all the data in a centralised, singular data warehouse can make it easier to keep the data secure compared to having to assess and monitor the security of many different databases. Overall, it provides a more centralised approach to data security.

- **Historical data**
  Storing historical data can be incredibly important for predictive analytics and studying trends over time and data warehouses allow this. The EMS could use an APDS data warehouse to analyse trends over time such as the rate of incidents per month or the locations which have had the most incidents in the past. Additionally, the predictive analytics components of historical data could allow the EMS to predict peak times and patient outcomes.
- **Legal and regulatory requirements**
  The centralised, all-in-one nature of data warehouses can make it easier for the APDS (and thus the EMS) to be compliant with legal and regulatory data requirements. This is because as a data warehouse is all one system, it can be easier to meet requirements as compared to having to meet requirements in many different systems.

Explain the Extract, Transform, Load (ETL) process and its significance in consolidating and integrating data from multiple sources.

The Extract, Transform, Load (ETL) process is used to combine data from various sources in an organisation into a data warehouse. I will now explain this ETL process in more detail.

Extract:
To perform the ETL process, the first step is to extract the data. This means extracting data from various data sources in an organisation such as databases, web services, files, etc. This data is extracted into the staging area (a temporary/interim storage area). The data is extracted to the staging area so that profiling and validating the data can be done before the data is processed into the data warehouse. Some profiling and validating that is done in the extraction phase can include identifying missing data, picking up inconsistencies in the data, etc.

Transformation:
The data extracted in the extract process is generally not ready to be used effectively in its raw form, hence the need for the transformation process. In the transformation process, validation functions are applied to the extracted data, typically in an effort to clean, structure, and enrich the data. This can involve data cleansing (removing duplicate values and fixing incorrect data), data structuring and integration (changing and merging formats of data), and data enrichment (adding metadata). There are *many* more transformation steps which could be done, it all depends on the organisation and their needs.

Load:
Loading the data is the final step of the ETL process. In this stage, the transformed data is loaded into the target system (likely a data warehouse in this case). Because of the large volume of data likely involved in the ETL process, the load process needs to be as efficient as possible so that it does not take more time than necessary. The type of loading in this process can depend on the amount of data as well as where the data is being loaded. Some loading types include:

- Initial load: data warehouse tables are populated
- Incremental load: new or modified data is loaded
- Streaming: data is loaded in real or close to real-time
- Full refresh: deleting data from certain tables in the data warehouse and reloading them with new data

Validations are also usually performed in the load stage as well. These validations can include checking that all the data is complete, checking the data integrity, testing tables, and overall general validation of the data and the data system.

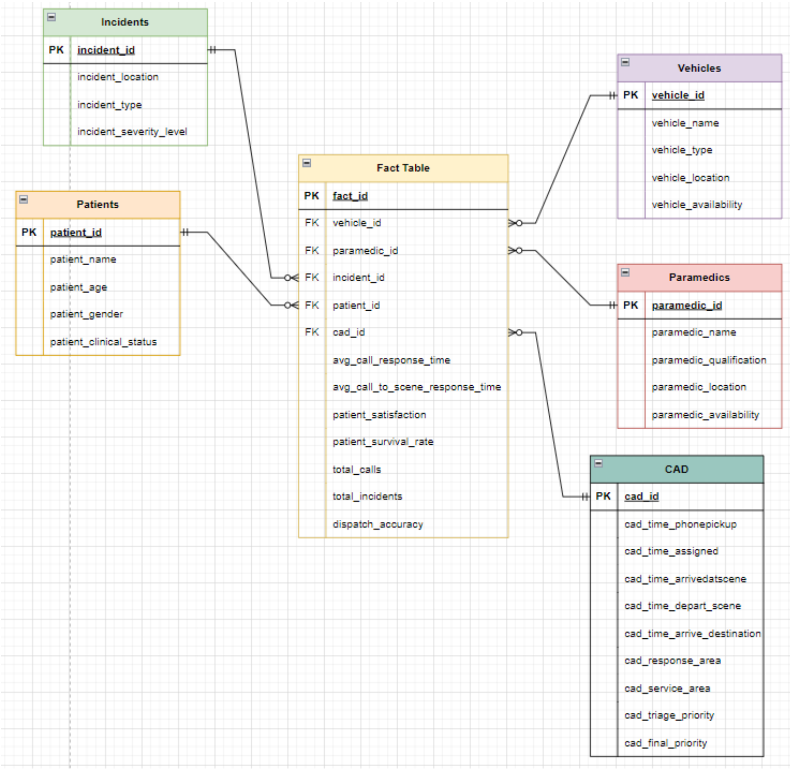
Significance:
ETL is significant and important in consolidating and integrating data from multiple sources. This is because it involves extracting data from various data sources in an organisation, then transforming this data to be compatible with the system it is being entered into, then transforming all the data into a common format/structure, then finally loading the data into a central data warehouse (or other kind of repository/system). The ETL process can also help greatly with maintaining/improving data quality and consistency. **Overall, the ETL process is very efficient, streamlined and functional, making it an effective strategy for organisations to follow to improve their data management**

Additional points on the ETL process being significant in consolidating and integrating data from multiple sources:

- Improving data quality: When data goes through the ETL process, it goes through a great amount of cleaning. Ultimately this profiling/validating/cleaning done throughout the ETL process (especially in the transformation stage) results in higher-quality data.
- Ensuring data compatibility: For data to be consolidated and integrated well, the data needs to be compatible with the data warehouse that it is being entered into. Using the ETL process, data compatibility can be accomplished and data can be successfully entered into the data warehouse.
- Enriching the data: In the transformation stage, the data can be enriched, ultimately increasing the value and usefulness of the data.
- Ensuring security: When the data goes through the ETL process, it can be checked against security rules and security validation measures to ensure that the data is secure and encrypted.
- Improving performance: Measures such as adding indexes and partitioning data can be implemented in the ETL process to improve performance.

Design a basic data warehouse schema for the Ambulance and Paramedic Dispatch System and describe the ETL process.

To design a basic data warehouse schema for the APDS, I will use a star schema. I will emphasise ‘basic’ here in that if the EMS were to create a data warehouse schema, it would likely be much more complex. This is just for demonstration purposes.



I will now describe the ETL process for the APDS:

Extracting the data:
To perform the ETL process for the APDS, the first step is to extract the relevant data from the various APDS/EMS data sources into the staging area (a temporary storage environment). In this context, this would likely mean extracting data relating to incidents, patients, vehicles, paramedics, etc. Once the data is in the staging area, validating and profiling can be performed to identify missing data, pick up inconsistencies, etc. This is done so that the data is ready for the transformation stage.

Transforming the data:
The extracted APDS would likely not be ready to be put into the data warehouse in its raw form, hence the need for the transformation process. Here are some steps that I would recommend for the EMS to transform the data.

Cleaning the data: This is similar to the profiling/validating aspect of the extract stage, but more in-depth. This can include removing duplicate data, ensuring consistency in the data, and correcting mistakes in the data. This is done so that the data is of high quality and is ready to be transferred into the data warehouse.

Integrating the data: This means merging the different APDS data into a singular, universal format. This is so it can be streamed into the data warehouse effectively and efficiently and so that it is easier to analyse the data overall.

Enriching the data: This means enriching the APDS data by adding other information to it such as metadata.

Validating the data: Using validation rules/processes to validate the APDS data to meet the needs and business rules of the EMS.

I have listed just a few aspects of the transformation process that the EMS may wish to involve, but they could utilise further transformation processes here as well.

Loading the data:
Once the APDS data has been successfully transformed, we can move on to the loading stage. In this stage, the transformed APDS data is loaded into the data warehouse. In this case, the data warehouse would be developed similarly to the basic data warehouse schema I have provided. Because the data warehouse is new and there is no data in it, the EMS should likely use an initial load to load the APDS data. They may also want to use incremental loads or other loading types depending on the circumstances. The EMS should also run validation processes in this stage such as checking that all the data is complete, checking the data integrity, testing tables, and overall general validation of the data and the data system.

Further considerations:
Once extracting, transforming, and loading the APDS data is finished, this does not mean that the ETL process for the EMS is entirely finished. The EMS should make sure to keep maintaining the system and altering it to fit new needs and requirements. They should also make sure to educate their team about the data warehouse so that they are not confused by it and are able to use it effectively.

Conclusion

Overall, this data management plan that I have developed for the EMS Ambulance and Paramedic Dispatch System should be greatly beneficial for the EMS in enabling them to continue to provide their important, life-saving care. I will now summarise the key findings and insights from this data management plan that I have developed:

Purpose of data management:
To begin this data management plan, I first delved into the purpose of data management in the context of the EMS case.

I first analysed the ambulance and paramedic dispatch system. I identified that the dispatch system was inefficient, resulting in delays in responding to critical emergencies. I also identified that poor resource allocation was hampering the efficiency of the care that the EMS delivers. I emphasised how, because of these issues, the EMS team were interested in a more capable dispatch system, and that they wanted an improved data management plan to do so.

I then identified and described the business requirements (dispatching efficiency and dispatching accuracy) and data requirements of the system. The data requirements included requirements such as (but not limited to) efficient and effective capturing, storing, and analysing of critical incident data, implementing an ETL plan, and establishing a data warehouse.

I then gave reasons for the importance of effective data management for the Ambulance and Paramedic Dispatch System. These reasons included (but were not limited to) better response times, optimised resource allocation, and increased trust and public image.

Database Design:
I created an accurate and appropriate ER diagram to meet the needs given by the EMS. I then, based on the requirements given, identified and defined the tables, relationships, and attributes which would be used. Finally, I clearly and thoroughly explained and justified why I made such decisions.

SQL Implementation:
I successfully implemented the previously created ER diagram into MySQL and gave reasons for why I used MySQL as my RDMS (security, reliability, speed, etc.). I wrote SQL scripts to create the required tables and populated each table with sample data. I then wrote SQL queries to demonstrate some useful functionalities such as:

- Retrieving incident locations and details
- Determining the quickest and alternative routes to an incident
- Retrieving information about premises and dangerous chemicals
- Generating detailed statistics on response times and callout causes

Data quality:
I identified many potential (and some current) data quality issues in the APDS. This included accuracy, volatility, completeness, timeliness, and consistency. To address these issues, I proposed actionable strategies for the EMS such as data validation, data cleansing, data integration, and governance practices. I then discussed the importance of data quality in maintaining an effective dispatch system. This included reasons such as response times, resource allocation, and legal/regulatory requirements.

Utilisation of data:
I discussed how data collected by the APDS can be combined with big data to enhance emergency response and prevention. I gave reasons including (but not limited to) preventative alerts, real-time tracking/monitoring, and public health management. I then provided examples of big data technologies and techniques that can be applied to the APDS such as NoSQL databases (MongoDB, Redis), machine learning (Apache Spark, TensorFlow), data visualisation (Tableau, Power BI) and MapReduce. Afterwards, I discussed the potential benefits and challenges of integrating big data with the existing system.

Benefits included:

- Reducing costs
- Faster response times
- Better real-time data capabilities
- Improved resource allocation
- Predicting future outcomes
- Better patient outcomes

Challenges included:

- Costs (costs can be either a benefit or a challenge)
- Data quality
- The integration process
- Issues with scalability
- Security
- Training

- Staff Unhappiness

Data Environment and Management Plan:
I comprehensively discussed data warehouse concepts including data integration and data warehouses being non-volatile, subject-oriented, and time-variant. I also discussed how data warehouses are relevant to the APDS, giving reasons such as data integration and centralisation, security, historical data, and legal and regulatory requirements. I then gave a clear explanation of the ETL process, covering the extract, transform, and load processes in great detail. I also emphasised the significance of consolidating and integrating data, giving reasons such as improving data quality, enriching the data, ensuring data compatibility, ensuring security, and improving performance. Finally, presented a thoroughly thought-out and well-design data warehouse schema for the APDS. I then explained the ETL process in the context of the APDS.

Final statement:
To conclude, I will say that the data management plan that I have provided should be highly effective in serving the EMS and their data management needs. It should address all the issues that the EMS is currently facing such as poor resource allocation and delays, as well as much more. If the EMS implements this data management plan as described, their quality of service should *significantly* improve, and thus, the EMS can save even more lives.