CS3101 P1 Report

Student ID: 240030041

Database Design for British Railway

This is a report for a database management system created for British Rail. It consists of an entity relationship diagram, translation into a relationship model, functional dependencies, normal form analysis, and minions assumptions.

Entity-Relationship Model

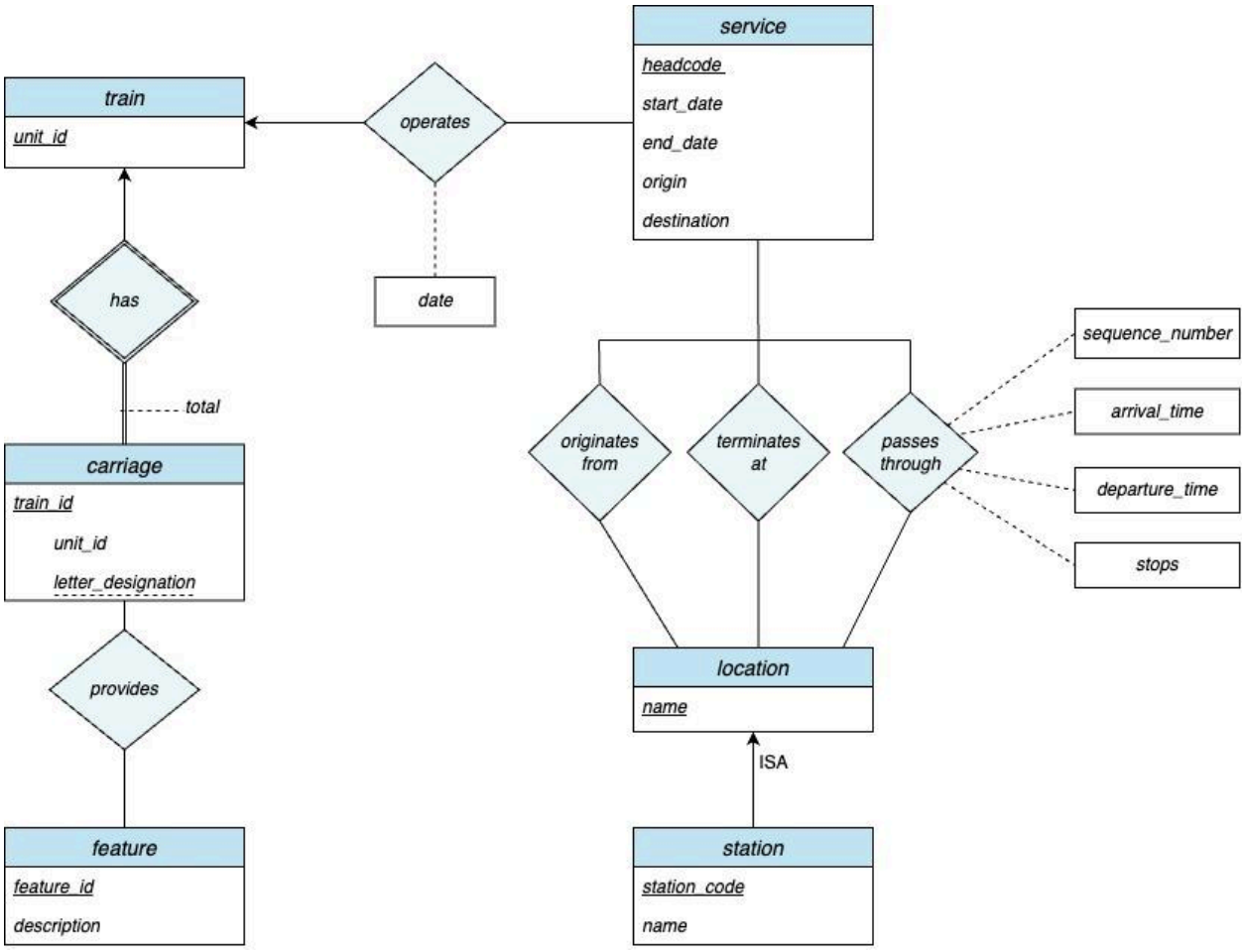

Figure 1 (Entity Relationship Diagram)

This ER model consists of the following entities and relationships:

Entities:

1. **service**: This entity represents a scheduled train journey that originates, passes through, and terminates at specific locations. It is identified by a unique *headcode*, valid between a start date and an end date. The *headcode* attribute is its primary key. It assumes that a service runs every day between its validity dates with the same train.

2. **location**:
   This entity represents all locations in the railway network. Each entity is identified by a unique name represented by the *name* attribute which is also its primary key. It assumes that each location has unique names and locations will not undergo name changes.

3. **station**: This is a specialised type of location. It inherits from the location entity. It is identified with a unique three-character code called *station_code*. All stations are locations, but not all locations are stations.

4. **train**: This entity represents a unit comprising a locomotive and one or more carriages. It is identified by a unique unit identifier represented by *unit_id*.

5. **carriage**: This entity represents an individual carriage within a train. It is a weak entity that is dependent on the strong train entity. It is identified by a letter designation within the context of the train which it is a part of, and represented by the key *train_id* which is a composite attribute of the *unit_id* (foreign key from train) and *letter_designation* (partial key).

6. **feature**: This entity represents various features and amenities within certain carriages. It is identified by the feature_id.

Relationships:

1. **train** *has* **carriage** (1-M): This is a one-to-many relationship between train and carriage. A train can have one or more carriages. Each carriage belongs to exactly one train.
2. **carriage** *provides* **feature** (M-N): This is a many-to-many relationship between carriage and feature. A carriage can have zero or many features, and a feature can be associated with 0 or many carriages.
3. **train** *operates* **service** (1-M): This is a one-to-many relationship between train and service. A train can operate one or more services, and a service is operated by exactly one train. This relationship has one attribute: *date*
4. **service** *originates from* **location** (M-1): This is a many-to-one relationship between service and location. It represents the origin of a service. A service has exactly one origin location, and a location can be the origin for multiple services.
5. **service** *terminates at* **location** (M-1): This is a many-to-one relationship between service and location. It represents the destination of a service. A service can have exactly one destination location, and a location can be the destination for multiple services.

6. **service** *passes through* **location** (M-N): This is a many-to-many relationship between service and location. It represents the various stops between the origin and destination of a service. A service passes through multiple locations in a sequence, and a location can be part of multiple services' routes. This relationship has attributes: *sequence_number, arrival_time, departure_time, and stops*
7. **location** ISA **station**: Station is a specialised type of location, distinguished by having a station code.

Conversion of ER Model to Relational Model

Relational Schemata

- **service** *(headcode, valid_from_date, valid_to_date, origin*, destination*, train_id*)
- **train** *(unit_id)*
- **carriage** *(train_id*, letter_designation*)
- **feature** *(feature_id, description)*
- **carriage_feature** *(train_id*, letter_designation*, feature_id*)
- **location** *(name)*
- **station** *(name*, station_code*)
- **service_location** *(headcode*, location_name*, sequence_number, departure_time, arrival_time, platform)

Functional Dependencies and Normal Forms

This design of this database attempts to minimise redundancy and improve data integrity. Using the relationship schema, the following functional dependencies have been identified for each relation (table). This model is in Boyce-Codd Normal Form (BCNF):

1. **location**: name → (no other attributes)
   This relation has only one attribute (name) which is the primary key, and there are no non-trivial functional dependencies, therefore it is in BCNF.

2. **station**: name → station_code; station_code → name
   This relation has two functional dependencies that form a cycle: *name* determines *station_code*, and *station_code* determines *name*. This means that both attributes are candidate keys and either can be used to identify a tuple. Therefore it satisfies BCNF's requirement that every determinant must be a superkey.

3. **train**: unit_id → (no other attributes)
   This relation only has one attribute, the unit_id attribute. It trivially satisfies BCNF as

there are no non-trivial functional dependencies possible.

4. **feature**: feature_id → description
   In this relation, *feature_id* functionally determines description. Since the primary key (*feature_id*) of this relation is a superkey, it satisfies the BCNF.

5. **service**: headcode → valid_from_date, valid_to_date, origin, destination, train_id
   In this relation, the *headcode* attribute functionally determines all other attributes in the relation. Since the primary key (*headcode*) of this relation is a superkey, it satisfies the BCNF.

6. **carriage**: (train_id, letter_designation) → (no other attributes)
   In this relation, the composite key (*train_id, letter_designation*) is the only determinant, so it's trivially in BCNF.

7. **carriage_feature**: (train_id, letter_designation, feature_id) → (no other attributes)
   This relation also entire consists of it's primary key attributes *(train_id, letter_designation, feature_id)*, therefore it trivially satisfies BCNF

8. **Service_location**:
   This relation has two candidate keys: (*headcode, location_name*) and (*headcode, sequence_number*). Each candidate key functionally determines all other attributes, therefore all determinants are superkeys. This relation satisfies BCNF.

Schedule Implementation: The concept of a schedule as defined as comprising "all services that depart or terminate at a given location". This is implemented through a database view instead of a separate relation. This view joins the *service* and *service_location* tables and can be filtered by the *location*. This was done to avoid data redundancy as the schedule information is derived from existing data. It also ensures that schedule information automatically reflects any updates to the service or location data.