Relational Schema:

RR_ACCIDENTS:

ACCIDNO: VARCHAR [PK], the unique code for a distinct accident

RAILROAD: VARCHAR [FK to RR_CLASS.RAILROAD], reporting railroad code

DATE: DATE, the date of the accident, YYYYMMDD

TRAIN_TYPE: CHAR(1), train type:

- F: freight train
- P: passenger train
- O: other train

ACC_TYPE: INT, accident type:

- 1=Derailment
- 2=Head on collision
- 3=Rearend collision
- 4=Side collision
- 5=Raking collision
- 6=Broken train collision
- 7=Hwy-rail crossing
- 8=RR Grade Crossing
- 9=Obstruction
- 10=Explosive-detonation
- 11=fire/violent rupture
- 12=Other impacts
- 13=Other

CARS: INT, number of hazmat cars involved in the accident

CARSDMG: INT, number of hazmat cars damaged or derailed in the accident

CARSHZD: INT, number of hazmat cars released in the accident

TOTAL_NUMBER_LOCO: INT, total number of locomotives in the train

TOTAL_LOCO_DERAIL: INT, total number of locomotives derailed in the train

TOTAL_NUMBER_CAR: INT, total number of cars in the train

TOTAL_CONSIST: INT, total number of vehicles in the train (cars + locomotives)

TOTAL_CAR_DERAIL: INT, total number of cars derailed in the accident

TOTAL_DERAIL: INT, total number of vehicles derailed in the accident

EVACUATE: INT, total number of persons evacuated

STATION: VARCHAR, nearest city or town

STATE_CODE: INT, FIPS state code

TEMP: INT, temperature in degrees Fahrenheit

VISIBLTY: INT, daylight period:

- 1: dawn
- 2: day
- 3: dusk
- 4: dark

WEATHER: INT, weather conditions:

- 1: clear
- 2: cloudy
- 3: rain
- 4: fog
- 5: sleet
- 6: snow

CAUSE: VARCHAR [FK to ACCIDENT_CAUSE.FRA_CAUSE_CODE], accident primary cause

CAUSE2: VARCHAR [FK to ACCIDENT_CAUSE.FRA_CAUSE_CODE], accident secondary cause, can be NULL

TRACK_TYPE: CHAR(2), accident track type:

- MS: mainline & siding track
- YI: yard & industry track

TRACK_CLASS: CHAR(1), accident track class

HIGHSPD: INT, train speed at accident

ACCDMG: INT, total reportable damage on all reports in $

LATITUDE: REAL, Latitude in decimal degrees, explicit decimal, explicit +/- (WGS84)

LONGITUD: REAL, Longitude in decimal degrees, explicit decimal, explicit +/- (WGS84)

STATE: VAR(2), state of the accident

TRAIN_WEIGHT: INT, train weight

RR_TRAFFIC:

TRAFFIC_CODE: VARCHAR [PK], the unique code for a distinct traffic entry

RAILROAD: VARCHAR [FK to RR_CLASS.RAILROAD], reporting railroad code

IYR: DATE, the year of the report, YYYY

IMO: DATE, the year of the report, MM

STATE: INT, FIPS state code, where the report was signed

COUNTY: VARCHAR, GSA county name, where the report was signed

YSMI: INT, yard switching miles for this month

FRTRNMI: INT, freight service miles for this month

PASTRNMI: INT, passenger service miles for this month

OTHERMI: INT, any other train miles not included in freight, passenger or yard switching

RR_INJURY:

REPORT_CODE: VARCHAR [PK], the unique code for a distinct injury report

DATE: DATE, the date of the incident, YYYYMMDD

RAILROAD: VARCHAR [FK to RR_CLASS.RAILROAD], reporting railroad code

TYPPERS: CHAR(1): type of person whose injury/illness is being reported:

- A= worker on duty-employee
- B= employee not on duty
- C= passenger on train
- D= non-trespassers-on railroad property
- E= trespassers
- F= worker on duty-contractor
- G= contractor-other
- H= worker on duty-volunteer
- I= volunteer-other

- J= non-trespassers-off railroad property

AGE: INT, age of person being reported

STATE: INT, FIPS state code

CASFATAL: CHAR(1), if fatality

COUNTY: VARCHAR, GSA county name

HZMEXPOS: CHAR(1), if hazmat exposure, Y, N, NA

LATITUDE: REAL, Latitude in decimal degrees, explicit decimal, explicit +/- (WGS84)

LONGITUD: REAL, Longitude in decimal degrees, explicit decimal, explicit +/- (WGS84)

RR_CLASS:

RAILROAD: VARCHAR, [PK], the unique code for a distinct railroad company

RAILROAD_NAME: VARCHAR, the full name for the railroad company

RAILROAD_SUCCESSOR: VARCHAR, the successor of this railroad company

RRCLASSIFICATION: INT, the railroad classification

ACCIDENTS_CAUSE:

FRA _CAUSE_CODE: CHAR(4), [PK], the unique code for a FRA accident cause code

ACCIDENT_CAUSE_CODE_DESCRIPTION: VARCHAR, accident cause description

ADL_CAUSE_SUBGROUP: VARCHAR, ADL cause group description

GROUP: CHAR(3), ADL cause group code

FRA_ACCIDENT_CAUSE_GROUP: VARCHAR, FRA cause category description

CATEGORY: CHAR(1), FRA cause category label

2. Explain your assumptions for each entity and relationship in your model. Discuss why you've modeled something as an entity rather than an attribute of another entity. Describe the cardinality of relationships, like why a student is linked to only one advisor. These assumptions might come from customer requirements or application constraints. Please clarify them.

The design of this project's database focuses on a railroad-company-oriented structure for documenting accidents, traffic, and injuries. It comprises five entities: RR_ACCIDENTS, RR_TRAFFIC, RR_INJURY, RR_CLASS, and ACCIDENT_CAUSE.

In RR_ACCIDENTS, each tuple represents a distinct railway accident reported exclusively by a single railway company. For RR_TRAFFIC, every tuple documents the varied types of mileage logged by a specific railroad company within a month. RR_INJURY contains individual injury reports, each filed by a railroad company. The common attribute 'RAILROAD' across RR_ACCIDENTS, RR_TRAFFIC, and RR_INJURY denotes the name of the reporting railroad company. The entity RR_CLASS contains detailed information about each railroad company, and entity ACCIDENT_CAUSE contains additional information on the railway accident cause group.

The relationships among entities are depicted in the ER diagram and further clarified here. RR_ACCIDENTS has a many-to-many relationship with ACCIDENT_CAUSE, as a single accident may have one or two causes. It also has a many-to-one relationship with RR_CLASS, since an accident is reported by only one railroad company, although a single company can report multiple accidents. RR_TRAFFIC is in a many-to-one relationship with RR_CLASS, where each traffic record is associated with a single railroad company, but a company can report multiple traffic volumes in RR_TRAFFIC for different months. Similarly, RR_INJURY maintains a many-to-one relationship with RR_CLASS, with each injury report linked to one railroad company, while a company can file multiple injury reports.

RR_ACCIDENTS, RR_TRAFFIC, and RR_INJURY are distinct entities due to their unique purposes in documenting railroad operations from three separate viewpoints. RR_CLASS is individualized because it contains additional information about each railroad company, such as its full name, successor, and classification, each uniquely defined. Lastly, ACCIDENT_CAUSE stands alone because it accommodates multiple methods for grouping railroad accident causes. While the default grouping method is the FRA cause code which is used in RR_ACCIDENTS, other grouping methods developed by researchers are also widely used in the industry and academia. This entity contains this additional information.

4. Describe why you choose to use BCNF vs 3NF,

We decide to use 3NF to normalize the database:

The complexity of the database design is simple and it is not complicated for us to find the relationship between attributes, and the performance of the database is a higher priority, and the slight redundancies that 3NF allows do not significantly impact the integrity of the data.

Our entities have been normalized to 3NF. By using candidate keys (ACCIDNO (PK of RR_ACCIDENTS, TRAFFIC_CODE (PK of RR_TRAFFIC), REPORT_CODE (PK of RR_INJURY), RAILROAD (PK of RR_CLASS), and FRA _CAUSE_CODE (PK of ACCIDENT_CAUSE), we can access all the attributes in all the entities.