# CIS 5500: Database and Information Systems
# Homework 2: Relational DB Design

February 26, 2026

Mustafa Rashid

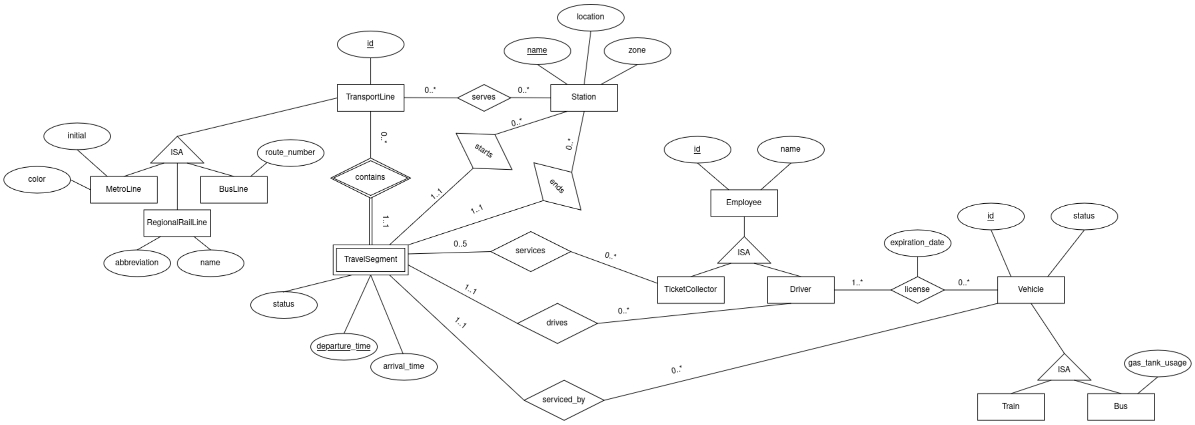
Spring 2026

---

color

1. **Question 1 (30 points)**

   A. **(20 points)**



   B. **(5 points)**

**Ans:** A TravelSegment could start or end at a Station that is not served by its associated TransportLine. This would be valid in the ER diagram because there is no constraint requiring the start and end stations of a TravelSegment to be among the stations served by that TransportLine.

To enforce this constraint, we must require that the start and end stations of a TravelSegment appear in the Serves relationship with the same TransportLine. In other words, a TravelSegment may only start and end at stations that are served by its associated TransportLine.

   C. **(5 points)**

**Ans:** Add an attribute to the relationship `Serves` called `no_of_platforms`. Add an attribute to the Station entity set called `maximum_no_of_platforms`.

We then add a constraint that, for each Station, the sum of `no_of_platforms` across all TransportLines assigned to that Station does not exceed `maximum_no_of_platforms`.

2. **Question 2 (20 points)**

   A. **12 points**

**Ans:**

```
CREATE TABLE Artists(
ArtistID INT PRIMARY KEY,
Name VARCHAR(50) NOT NULL,
Nationality VARCHAR(50) NOT NULL,
BirthYear INT(4) NOT NULL
);

CREATE TABLE Customers(
CustomerID INT PRIMARY KEY,
Name VARCHAR(50) NOT NULL,
Email VARCHAR(100) NOT NULL
);

CREATE TABLE Artworks(
ArtworkID INT PRIMARY KEY,
OwnerID INT FOREIGN KEY REFERENCES Customers(CustomerID),
CreatorID INT FOREIGN KEY REFERENCES Artists(ArtistID),
Title VARCHAR(50) NOT NULL,
AYear INT(4),
Medium VARCHAR(50)
);

CREATE TABLE Exhibitions(
ExhibitionID INT PRIMARY KEY,
ExhibitionName VARCHAR(100) NOT NULL
);

CREATE TABLE DisplayedIn(
ArtworkID INT,
ExhibitionID INT,
PRIMARY KEY (ArtworkID, ExhibitionID),
FOREIGN KEY (ArtworkID) REFERENCES Artworks(ArtworkID),
FOREIGN KEY (ExhibitionID) REFERENCES
   Exhibitions(ExhibitionID),
StartDate DATE,
EndDate DATE
);
```

B. **4 points**

**Ans:** We can create an assertion as follows

```
        CREATE ASSERTION exhibition_cardinality
CHECK (
    NOT EXISTS (
        SELECT ExhibitionID
        FROM DisplayedIn
        GROUP BY ExhibitionID
        HAVING COUNT(*) NOT BETWEEN 5 AND 20
    )
);
```

C. **4 points**

**Ans:** We change the cardinality on the Artists entity set from 0..∗ to 1..∗. We also add the following assertion

```
CREATE ASSERTION artist_has_artwork
CHECK (
  NOT EXISTS (
    SELECT *
    FROM Artists A
    WHERE NOT EXISTS (
      SELECT *
      FROM Artworks W
      WHERE W.CreatorID = A.ArtistID
    )
  )
);
```

3. **Question 3 (40 points)**

   A. **4 points**

   **Ans:** No they cannot.
   If they did then the functional dependencies PassengerID → PassengerEmail and PassengerEmail → PassengerID would not hold.

   B. **4 points**

   **Ans:** Yes. FINISH ME!

   C. **6 points**

   **Ans:** passengerID,trainNO and passengerEmail, TrainNO FINISH ME!

   D. **5 points**

   **Ans:** FINISH ME!

   E. **2 points**

   **Ans:** FINISH ME!

   F. **2 points**

   **Ans:** FINISH ME!

   G. **12 points**

   **Ans:** FINISH ME!

   H. **5 points**

   **Ans:** FINISH ME!