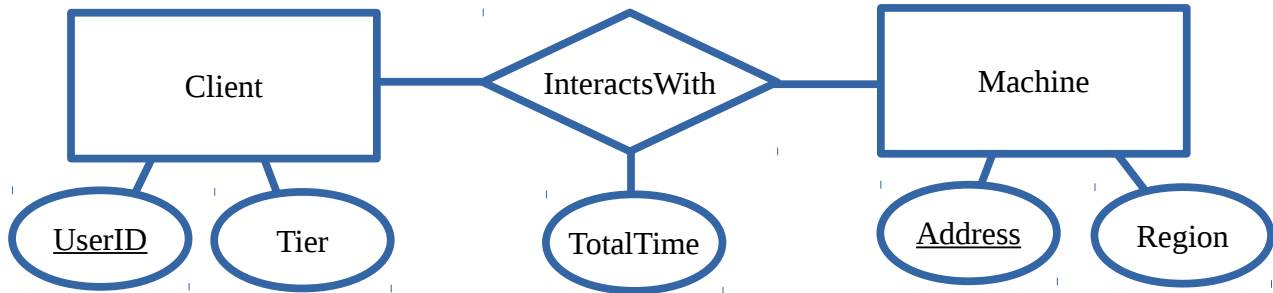


## Section 7 Worksheet

ER → SQL

Coming up with ER diagrams is relatively simple compared to coming up with a representation in SQL. In this part we will try to convert some ER diagrams into SQL CREATE TABLE statements. **For the following ER Diagrams and respective specifications, come up with the appropriate schema and implement it in SQL.** Feel free to abbreviate names to save time and assume attribute types as appropriate.



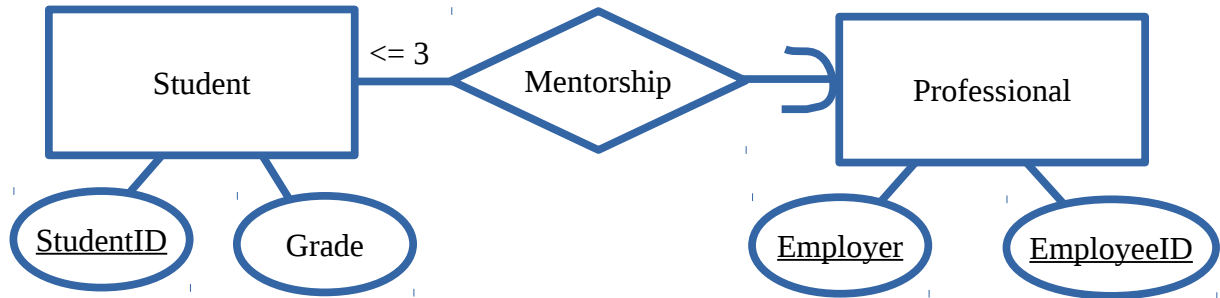
Other Specification:

If a Client is removed, we want to retain the interaction history.

If a Machine is removed, we don't want to retain the interaction history.

Any updates for a Client or Machine should also update respective interaction records.

```
CREATE TABLE Client (
  UserID INT PRIMARY KEY,
  Tier VARCHAR(20)
);
CREATE TABLE Machine (
  Address VARCHAR(40) PRIMARY KEY,
  Region VARCHAR(50)
);
CREATE TABLE InteractsWith (
  TotalTime FLOAT,
  UserID INT REFERENCES Client
    ON UPDATE CASCADE
    ON DELETE SET NULL,
  Address VARCHAR(40) REFERENCES Machine
    ON UPDATE CASCADE
    ON DELETE CASCADE,
  PRIMARY KEY (UserID, Address)
);
```



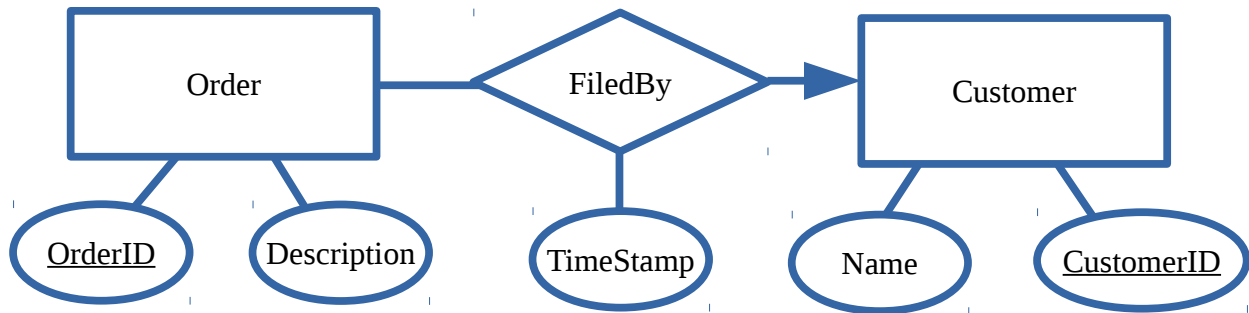
Other Specification:

Only Students that are grade 8 or higher are allowed to have mentorships.

Any updates for a Student or Professional should also update respective interaction records.

```

CREATE TABLE Student (
  StudentID VARCHAR(50) PRIMARY KEY,
  Grade INT CHECK (Grade >= 8),
  MentorEmployer VARCHAR(100) NOT NULL,
  MentorID VARCHAR(50) NOT NULL,
  FOREIGN KEY (MentorEmployer, MentorID) REFERENCES Professional(Employer, EmployeeID)
  ON UPDATE CASCADE
);
CREATE TABLE Professional (
  Employer VARCHAR(100),
  EmployeeID VARCHAR(50),
  PRIMARY KEY (Employer, EmployeeID)
);
CREATE ASSERTION MyAssert CHECK (
  3 >= ALL (SELECT count(*)
            FROM Student AS S
            GROUP BY S.MentorEmployer, S.MentorID)
);
-- alternatively you could use a trigger
CREATE TRIGGER MyTrigger
ON Student
AFTER INSERT
AS IF (3 < ANY (SELECT count(*)
                FROM Student AS S
                GROUP BY S.MentorEmployer, S.MentorID))
ROLLBACK TRANSACTION;
  
```



Other Specification:

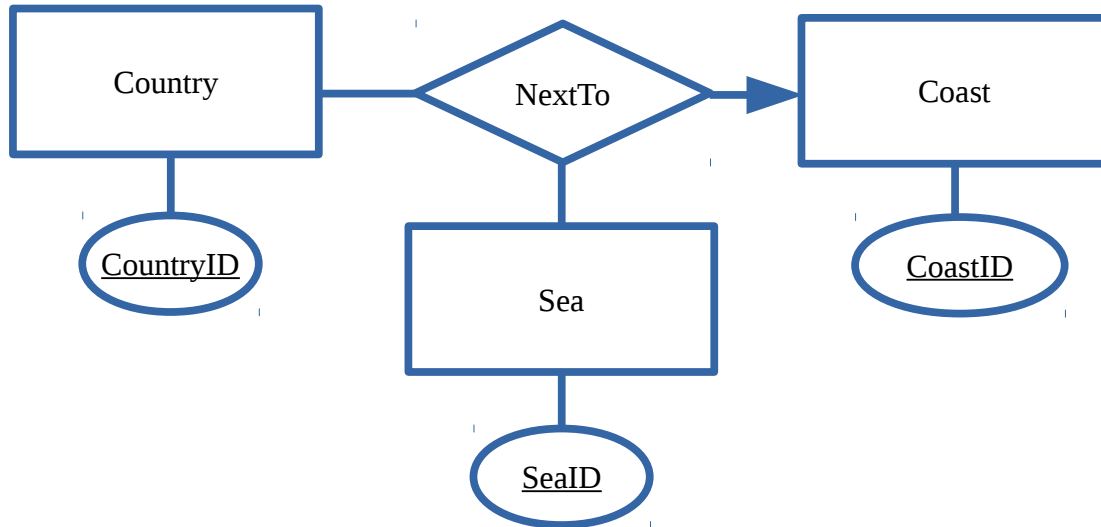
If a customer is removed we should have all respective orders altered so they are from the default anonymous user with CustomerID 0.

Order descriptions should be sufficiently specific, so we need to make sure the description is at least 25 characters long.

Any updates for an Order or Customer should also update respective interaction records.

```

CREATE TABLE Order (
  OrderID INT PRIMARY KEY,
  Description VARCHAR(255) CHECK (len(Description) >= 25),
  TimeStamp VARCHAR(50),
  CustomerID INT REFERENCES Customer
  ON UPDATE CASCADE
  ON DELETE SET DEFAULT
);
CREATE TABLE Customer (
  Name VARCHAR(100),
  CustomerID INT DEFAULT 0 PRIMARY KEY,
);
  
```



Other Specification:

Any updates for an Country, Sea, or Coast should also update respective interaction records.

```

CREATE TABLE Country (
  CountryID INT PRIMARY KEY
);
CREATE TABLE Sea (
  SeaID INT PRIMARY KEY
);
CREATE TABLE Coast (
  CoastID INT PRIMARY KEY
);
CREATE TABLE NextTo (
  CountryID INT REFERENCES Country
  ON UPDATE CASCADE,
  SeaID INT REFERENCES Sea
  ON UPDATE CASCADE,
  CoastID INT REFERENCES Coast
  ON UPDATE CASCADE,
  PRIMARY KEY (CountryID, SeaID)
);
  
```