**HW #2**

Akash Ghosh

**6.1E :**

R (ISBN, title, author(name, date

Of birth), publisher(name,

address(streetnr, streetname, zipcode, city)), pages, price)

Functional Dependencies Identification :

* ISBN > Title, pages, price (Each book has unique ISBN number)
* ISBN > PublisherName ( Each book has only one publisher)
* PublisherName> StreetNR, StreetName, ZipCode, City (Each publisher has unique addresses)
* Authorname > DateOfBirth (Each author has unique date of birth and name)

1NF Normalization:

Removing multivalued attributes.

* Author (name , date\_of\_birth)
* Publisher (address)

Result after 1NF normalization :

1. Book (ISBN, Title, Pages, Price, PublisherName)
2. Author (AuthorName, DateOfBirth)
3. Publisher(PublisherName, streetNr, StreetName, ZipCode,City)
4. Book\_contributor (ISBN, AuthorName)

2NF Normalization: Tables already satisfying 2NF form. No need to do the 2NF form since there are no partial dependencies.

3NF Normalization:

* Book (ISBN(PK), pages)
* Author (AuthorName(pk), DateOfBirth)
* Publisher (PublisherName(PK), StreetNr,StreetName, Zipcode, City)
* Book\_contributor(ISBN(FK),AuthorName(FK)
* Book\_Publisher(ISBN(Pk)(FK), publisherName(pk)(fk))

If a book have multiple publisher then relationship between book and publisher change to many to many relationship.

Book\_publisher(ISBN, PublisherName, NumberofCopies,Price)

Since price can vary by publisher to publisher, we moved the table book to book\_publisher.

**6.3E** :

While it offers a high-level conceptual representation of data, the Enhanced Entity-Relationship (EER) model is not always effective in enforcing business rules and constraints. The following are some instances of semantics that this EER model is unable to enforce:

**Business Rules on Contract Types**

The model includes an attribute Contract Type for **Customers**, but it does not enforce rules such as:

* A **Business customer** must have a different contract structure than a Residential customer.
* Certain **contract types** may only be available to specific types of customers.

**Customer-Reseller Contract Constraints**

The relationship Has\_Contract\_With between CUSTOMER and ELECTRICITY RESELLER does not enforce:

* + A customer cannot have multiple contracts with different resellers at the same time.
  + A business customer must have a VAT number, but no enforcement exists.

Mapping the EER model to a Relational Model:

* CREATE TABLE Electricity\_Producer

( Name VARCHAR(255) PRIMARY KEY, Location VARCHAR(255) NOT NULL );

* CREATE TABLE Electricity\_Reseller ( Name VARCHAR(255) PRIMARY KEY, Brand\_Name VARCHAR(255), Manager VARCHAR(255),

Number\_Of\_Employees INT CHECK (Number\_Of\_Employees > 0) );

* CREATE TABLE Customer (

ID INT PRIMARY KEY,

Contract\_Type VARCHAR(50) NOT NULL

);

* CREATE TABLE Residential ( ID INT PRIMARY KEY, Household\_Size INT CHECK (Household\_Size > 0), FOREIGN KEY (ID) REFERENCES Customer(ID);
* CREATE TABLE Business ( ID INT PRIMARY KEY, VAT\_Number VARCHAR(50) UNIQUE NOT NULL, FOREIGN KEY (ID) REFERENCES Customer(ID);
* CREATE TABLE Reseller\_Customer\_Contract ( Customer\_ID INT, Reseller\_Name VARCHAR(255), PRIMARY KEY (Customer\_ID, Reseller\_Name), FOREIGN KEY (Customer\_ID) REFERENCES Customer(ID) ;
* CREATE TABLE Power\_Plant ( ID INT PRIMARY KEY, Location VARCHAR(255) NOT NULL, MW\_H DECIMAL(10,2) NOT NULL );
* CREATE TABLE Producer\_Owns\_PowerPlant ( Producer\_Name VARCHAR(255), PowerPlant\_ID INT, PRIMARY KEY (Producer\_Name, PowerPlant\_ID), FOREIGN KEY (Producer\_Name) REFERENCES Electricity\_Producer(Name);
* CREATE TABLE Local\_Regulator ( Name VARCHAR(255) PRIMARY KEY, Province VARCHAR(255) NOT NULL, Manager VARCHAR(255) NOT NULL );
* CREATE TABLE PowerPlant\_Regulation ( Regulator\_Name VARCHAR(255), PowerPlant\_ID INT, PRIMARY KEY (Regulator\_Name, PowerPlant\_ID), FOREIGN KEY (Regulator\_Name) REFERENCES Local\_Regulator(Name);
* CREATE TABLE Gas\_Plant ( PowerPlant\_ID INT PRIMARY KEY, Number\_Of\_Pipelines INT CHECK (Number\_Of\_Pipelines > 0), FOREIGN KEY (PowerPlant\_ID) REFERENCES Power\_Plant(ID) ON DELETE CASCADE );
* CREATE TABLE Wind\_Plant ( PowerPlant\_ID INT PRIMARY KEY, Number\_Of\_Mills INT CHECK (Number\_Of\_Mills > 0), FOREIGN KEY (PowerPlant\_ID) REFERENCES Power\_Plant(ID) ON DELETE CASCADE );
* CREATE TABLE Nuclear\_Plant ( PowerPlant\_ID INT PRIMARY KEY, Number\_Of\_Reactors INT CHECK (Number\_Of\_Reactors > 0), FOREIGN KEY (PowerPlant\_ID) REFERENCES Power\_Plant(ID) ON DELETE CASCADE );

**6.4E**

The Enhanced Entity-Relationship (EER) model provides a conceptual foundation for building data interactions, but it is unable to enforce specific business rules and restrictions. Some instances of these constraints are:

**Employee Specialization Constraints (Pilot, Steward, and Flight Planner)**

* The EER model allows one person to hold many roles at the same time.  
  The approach does not require Pilots to have specific training or qualifications.  
  A Steward(ess) must meet demands.

**Airport and Airplane Ownership**

* Although the AIRPORT has staff, it does not follow strict restrictions like as limiting the number of planes per day.  
  A plane can only belong to one airport at a time.

**Ticket and Flight Relationship**

Although allows a ticket to be valid for multiple flights, but it does not stop:

* A ticket being valid for flights at conflicting times.
* A customer purchasing a ticket for an already full flight.

EER Model to Relational Database:

* CREATE TABLE Airport ( Name VARCHAR(255) PRIMARY KEY, Three\_Letter\_Abbreviation CHAR(3) UNIQUE NOT NULL, Location VARCHAR(255) NOT NULL );
* CREATE TABLE Employee (

ID INT PRIMARY KEY,

Home\_Address VARCHAR(255) NOT NULL

);

* CREATE TABLE Pilot ( ID INT PRIMARY KEY, Years\_Of\_Experience INT CHECK (Years\_Of\_Experience >= 0), FOREIGN KEY (ID) REFERENCES Employee(ID);
* CREATE TABLE Stewardess ( ID INT PRIMARY KEY, FOREIGN KEY (ID) REFERENCES Employee(ID)
* CREATE TABLE Flight\_Planner ( ID INT PRIMARY KEY, FOREIGN KEY (ID) REFERENCES Employee(ID);
* CREATE TABLE Customer ( ID INT PRIMARY KEY, Name VARCHAR(255) NOT NULL, Date\_Of\_Birth DATE NOT NULL, Number\_Of\_AirMiles INT CHECK (Number\_Of\_AirMiles >= 0) ;
* CREATE TABLE Plane ( ID INT PRIMARY KEY, Year\_Of\_Production INT NOT NULL, Type VARCHAR(50) NOT NULL, Number\_Of\_Seats INT CHECK (Number\_Of\_Seats > 0) NOT NULL );
* CREATE TABLE Flight ( ID INT PRIMARY KEY, Departure VARCHAR(255) NOT NULL, Destination VARCHAR(255) NOT NULL, Plane\_ID INT NOT NULL, FOREIGN KEY (Plane\_ID) REFERENCES Plane(ID);
* CREATE TABLE Ticket ( ID INT PRIMARY KEY, Departure VARCHAR(255) NOT NULL, Destination VARCHAR(255) NOT NULL, Price DECIMAL(10,2) NOT NULL );
* CREATE TABLE Customer\_Buys\_Ticket ( Customer\_ID INT, Ticket\_ID INT, PRIMARY KEY (Customer\_ID, Ticket\_ID), FOREIGN KEY (Customer\_ID) REFERENCES Customer(ID) ON DELETE CASCADE, FOREIGN KEY (Ticket\_ID) REFERENCES Ticket(ID);
* CREATE TABLE Ticket\_Flight\_Validity ( Ticket\_ID INT, Flight\_ID INT, PRIMARY KEY (Ticket\_ID, Flight\_ID), FOREIGN KEY (Ticket\_ID) REFERENCES Ticket(ID) ON DELETE CASCADE, FOREIGN KEY (Flight\_ID) REFERENCES Flight(ID) ;

Loss of Semantics in the Relational Model:

**Cardinality of Employee Assignments**

* The model does not enforce that a Pilot cannot be assigned to multiple flights at the same time.
* Possible Fix: Use a time constraint to prevent overlapping assignments.

**Ticket Validity and Flight Overbooking**

* The schema allows a ticket to be valid for multiple flights, but it does not prevent a flight from exceeding its seat capacity.