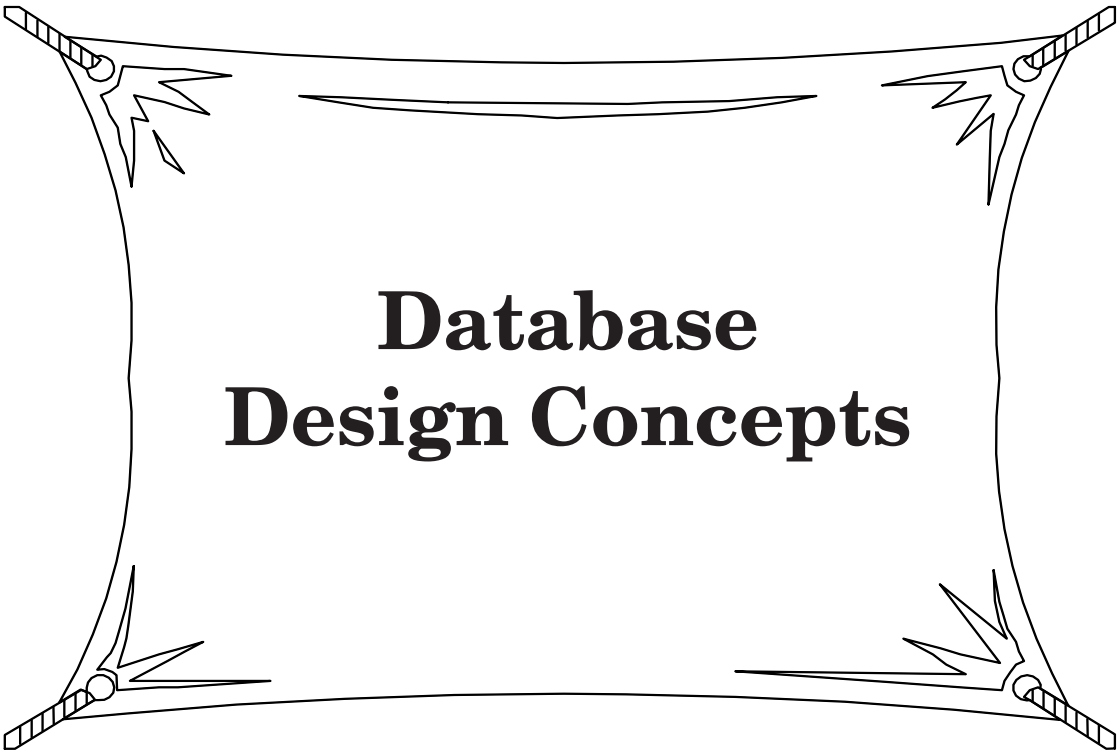


Database Design Concepts

Objectives

- become familiar with the terminology associated with a proper logical data base design

- based on user requirements, identify the key entities that satisfy the business requirements of the user

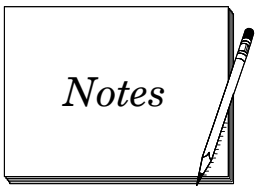
- create entity relationship diagrams

- become familiar with the rules and rigors of data normalization

- associate data elements with their corresponding entities (*data normalization*)

- verify proper primary and foreign keys in the logical data base design

- explain and utilize the concepts of generalization, supertypes and subtypes



Design Terminology

Data Modeling

Normalization

Logical Design

Data Modeling: The process of determining the data elements/information to be stored and how they will be used in an application or organization.

Logical Design: How you would **LIKE** to structure your data without regard to hardware, database manager, budget or performance requirements. **Focus is on following the rules of relational data.**

Physical Design: How you **HAVE** to structure your data based on hardware, database manager, budget and performance requirements. Focus is on how the data is used.

Normalization: A process for performing logical design.

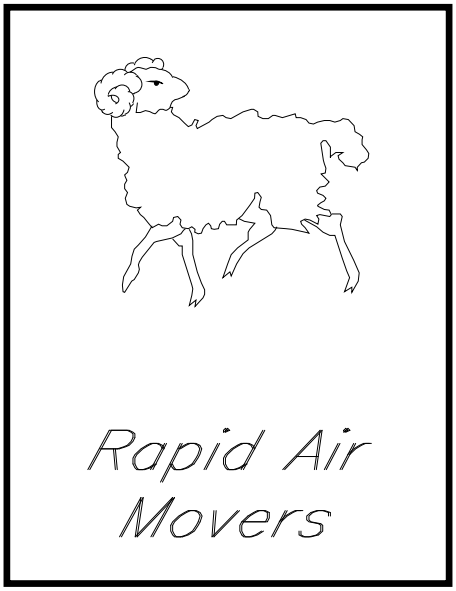
Identify the main entities

Designate Primary keys

Establish relationship between entities

Populate entities with data elements

Verify proper primary/foreign keys



Required Data Elements:

Passenger Name

Passenger Address

Passenger Telephone

Passenger Identifier (assigned by airline)

Seat Number

Fare

Airport City Codes

Airport City Name and State

Airport City Elevation

Meal Designation (can vary for each flight)

Flight Number

Arrival and Departure Cities

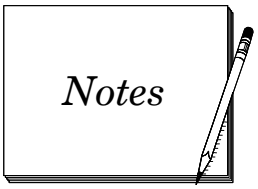
Alternative Landing Cities and Mileage (3)

Flight Mileage

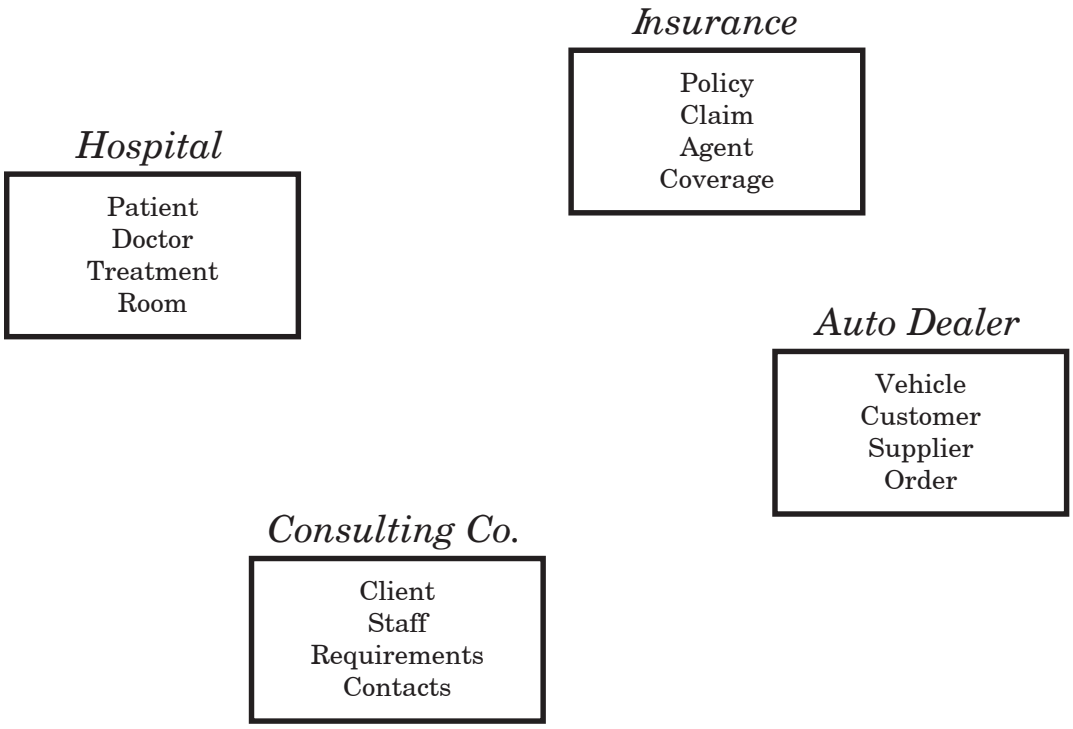
Frequent Flier Mileage Accrued

Smoking Passenger Indicator

Flight Date and Time



Entities



Identify the main entities

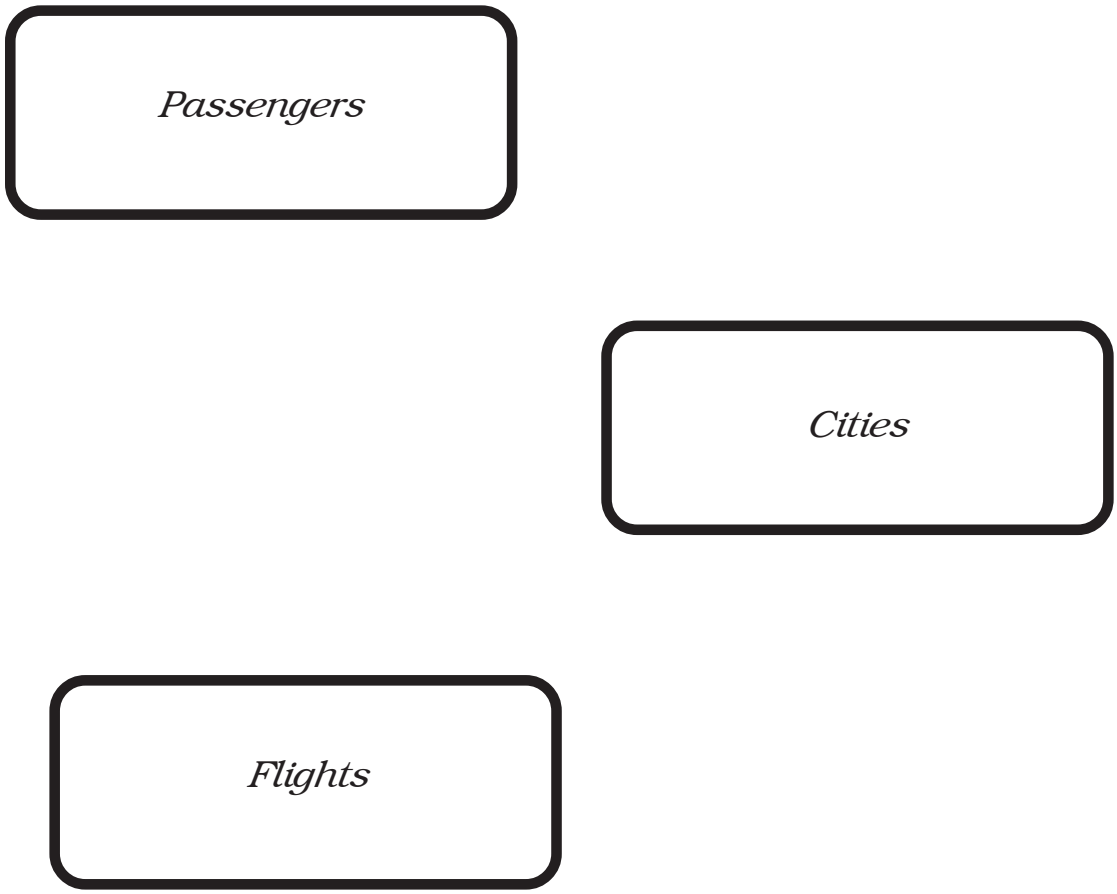
entity: an object, event, or transaction for which there are one or more *attributes*

attribute: a characteristic or property associated with an entity.

- *entities* become *tables*
- *attributes* become *columns* of tables

Entities

In our system...



Primary Keys

Designate Primary keys

What is a primary key?

A unique identifier. A primary key may be a data element, or a concatenation of data elements.

Passengers

Primary Key: (Passenger Identifier)

Assumption: No two passengers can have the same identifier

Primary Key: (City Code)

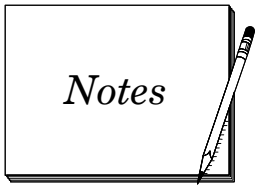
Assumption: No two cities can have the same city code

Cities

Flights

Primary Key: (Flight No. + Dept Date)

Assumption: Each flight will have its daily unique flight number assigned



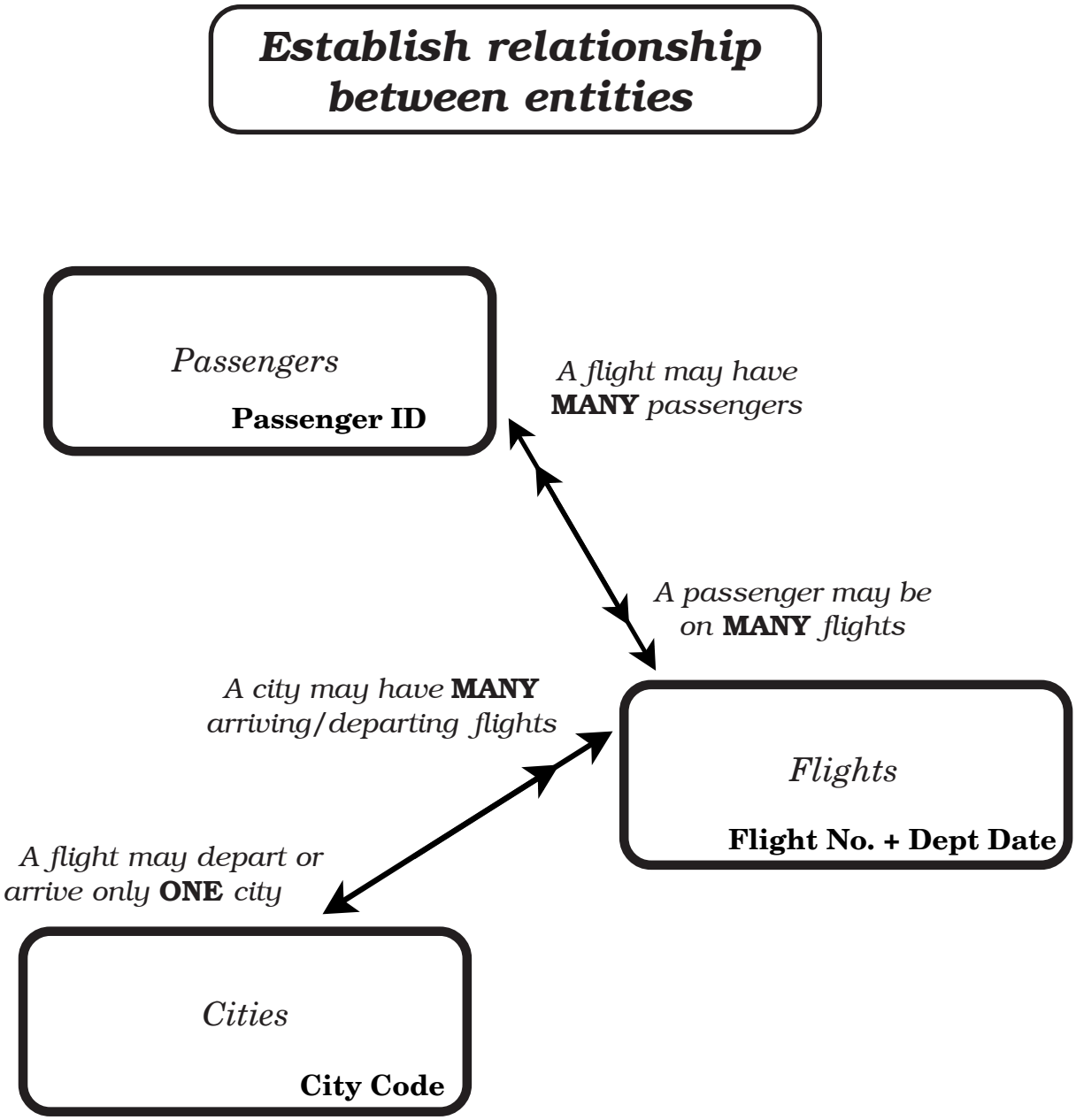
Entity Relationships

Relationship	*Definition*	*How to Handle*
One-to-One	*For each instance in an entity (parent), instances in the other entity (dependent) can occur only once*	**Combine into a single entity**
One-to-Many	*For each instance in an entity (parent), instances in the other entity (dependent) can occur many times*	**Goal in Relational Logical Design - Do Nothing!**
Many-to-Many	*For each instance in **either** entity (parent or dependent), instances in the other entity can occur many times*	**Break into a pair of one-to-many relationships; create a new entity which contains the primary keys of the original entities**

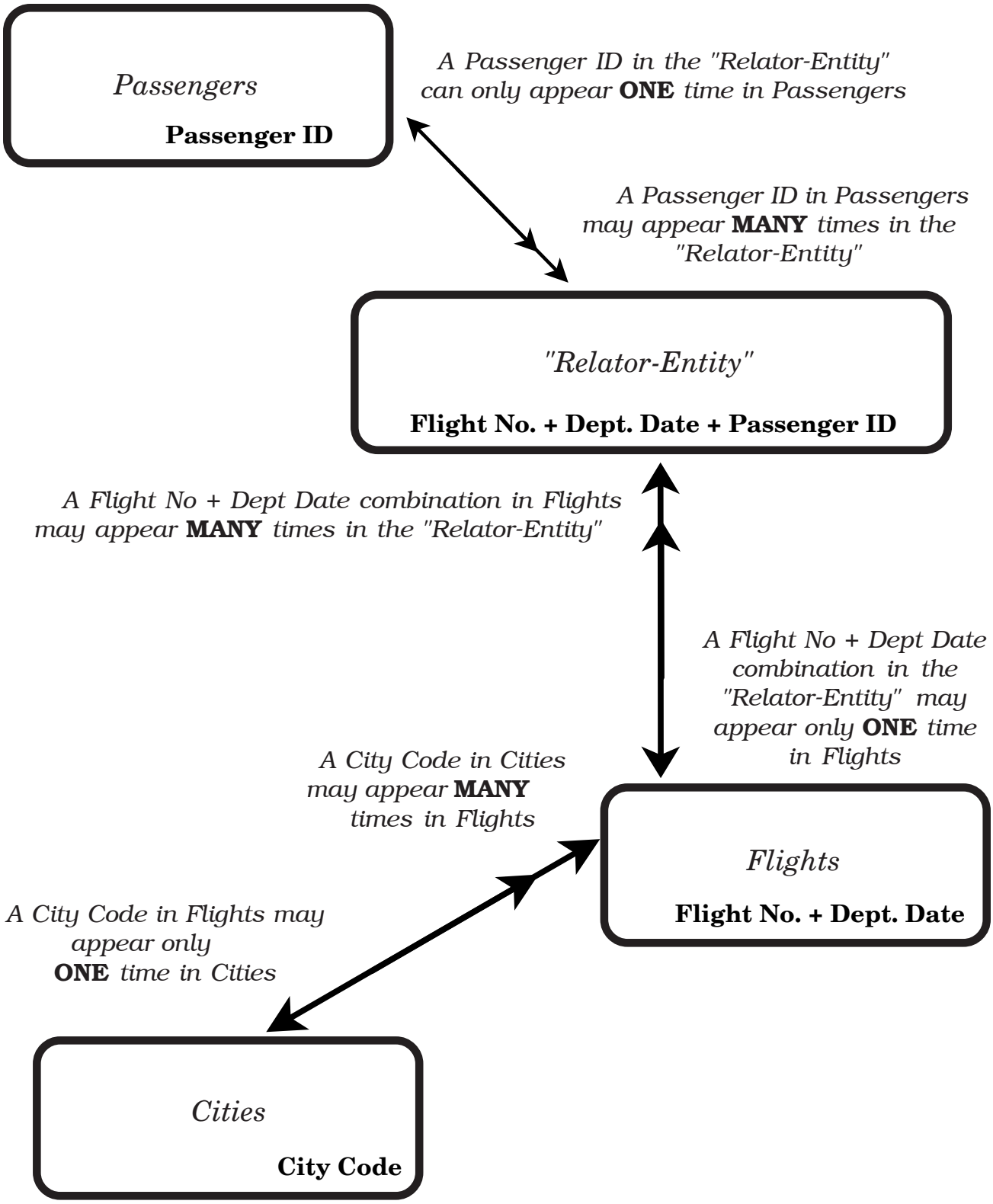
Cardinality: the ratio of instances of a parent to the instances of a dependent; the number of instances in an entity.

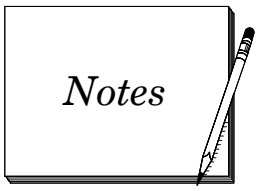
Foreign key: the attribute(s) in a dependent entity instance that exactly match the primary key of an existing instance in the parent entity.

Entity Diagram



Entity Diagram





Populate entities with data elements

The Goal: Third Normal Form

First Normal Form:

For each entity, create a separate entity for repeating data groups.

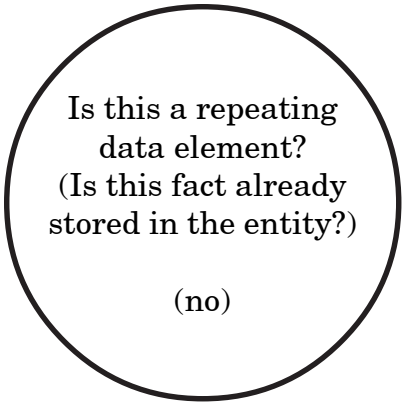
Second Normal Form:

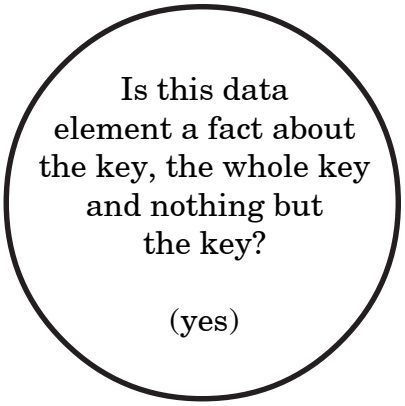
For each entity, create a separate entity for data elements that are based on a portion of the primary key.

Third Normal Form:

For each entity, create a separate entity for data elements that are based on some other data element.

Guidelines:

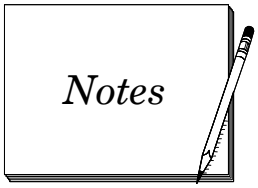




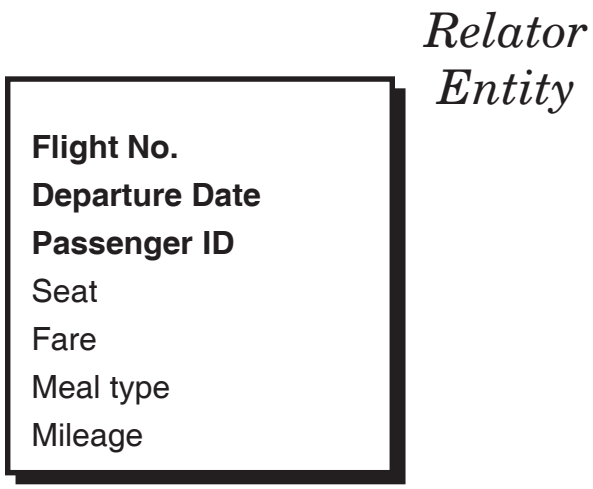
If data element does not "belong" in the entity you have placed it, you should either:

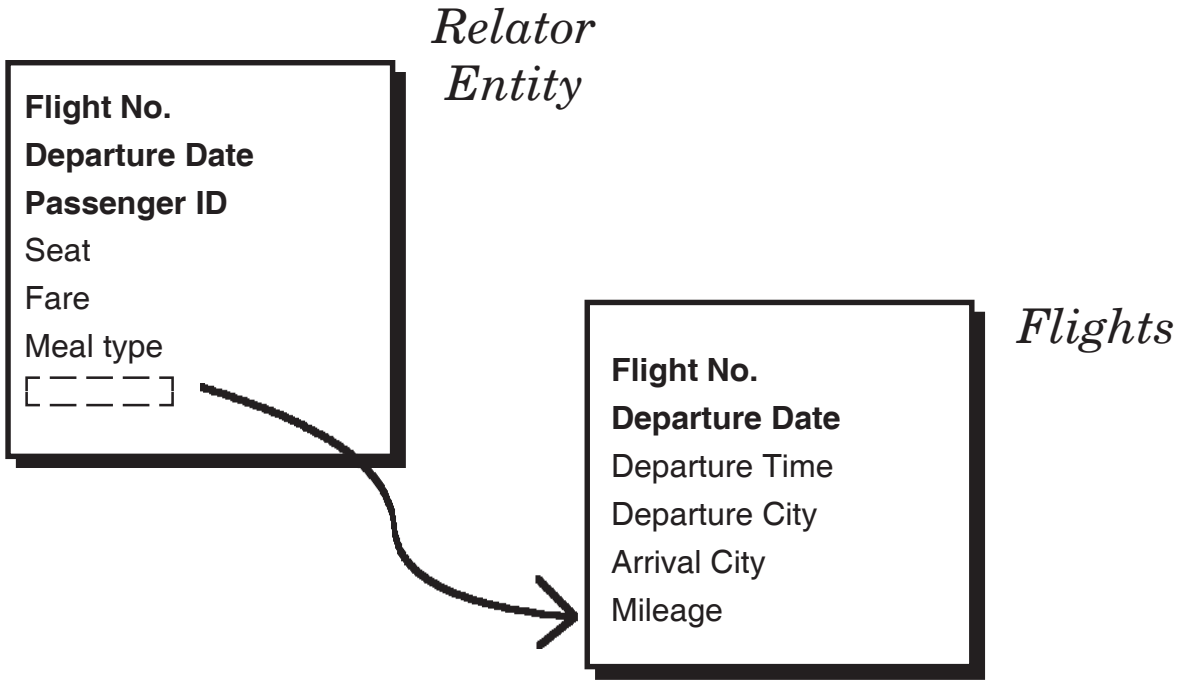
1. Find an existing entity in which it belongs and place it there.

2. Create a **new** entity for the data element.

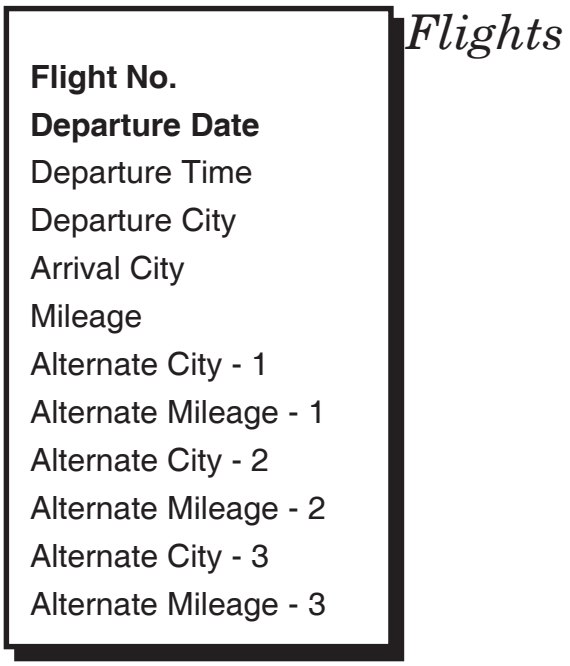


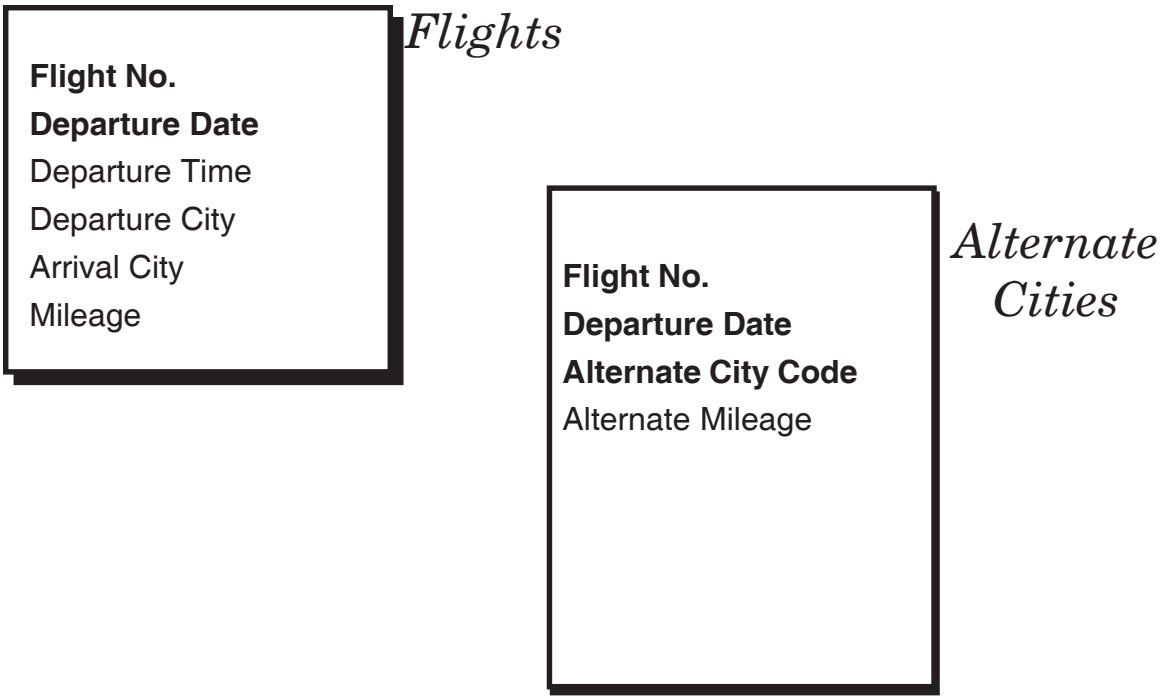
Data Normalization Example-1





Data Normalization Example-2





Data Normalization Result

Passengers
Passenger ID
Name
Addr
Smoker Ind
Phone
FF Miles

Passenger ids must be unique
1 phone number/address per passenger
1 frequent flyer account per passenger

Reservations
Flight No
Dept Date
Passenger ID
Seat
Fare
Meal Type

1 seat per reservation
1 reservation per passenger
Fare determined at time of reservation

Flights
Flight No
Dept Date
Dept Time
Dept City
Arrival City
Mileage

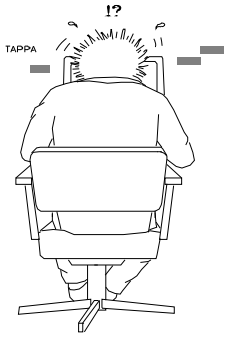
Departure information may differ
for same flight number on another date

Cities
City Code
City Name
State
Elevation

Alt. Cities
Flight No
Dept Date
City Code
City Mileage

A flight may have any number of
alternate cities on any given date

Why Normalize?



- Very good design methodology for relational model
- Easy, direct translation from logical design to physical design to implementation in true relational data base management systems
- Minimize data redundancy
- Avoid potential update and delete problems
- Provides a simple vehicle to add, delete or modify entities, attributes and relations without major restructuring of the tables or application
- Smaller tables, smaller rows and fewer total bytes per row
- Tables with fewer columns and shorter rows allows more related data to be stored together which reduces I/O

Verify proper primary/foreign keys

**Foreign key must exactly match the primary key:
same number of columns, same order of columns and same content**

Foreign key columns do NOT necessarily have to also be primary key columns of their own tables.