Table Relations

Database Design and Rules

greenwich.edu.vn

UNIVERSITY of GREENWICH

Alliance with FPT Education

Table of Content

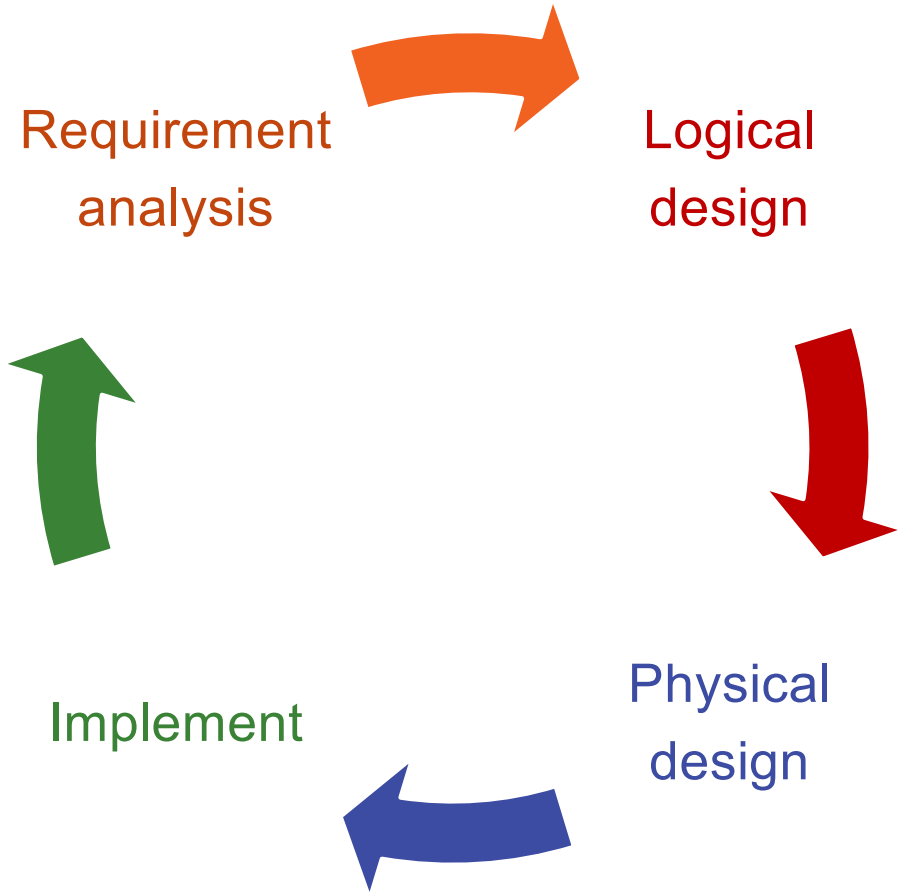
1. Database Design
2. Table Relations
3. Cascade Operations
4. E/R Diagrams

DATABASE DESIGN

Fundamental Concepts

Database life cycle

- Requirement analysis: study the data required for processing, the natural data relationships, and the software platform for the database implementation
- Logical design: create a conceptual data model diagram that shows all the data and their relationships.
- Physical design: select access methods, partitioning and clustering of data
- Implementation: create the database using the data definition language (DDL) and manipulate data with data manipulation language (DML)



LOGICAL DESIGN

Data modelling

- An integrated collection of concepts for describing and manipulating
  - Data
  - Relationship between data
  - Constraints on the data
- It is the representation of real-world objects and events, and their associations
- A data model comprises three components
  - Structural part: set of rules to construct the DB
  - Manipulative part: defining types of allowed operations
  - Set of integrity constraints: to ensures data is accurate

Data models

- It is quite helpful to the database designer who must communicate with end users about their data requirements
- It describes, in diagram form, the entities, attributes, and relationships that occur in the system
- There are many approach methods:
  - Object-based data models
  - Record-Based Data Models
  - Relational Data Model
  - Network Data Model
  - Hierarchical Data Model

Entity-Relationship (ER) model

- ER model is a kind of object-based data model
- Provide an intuitive image about different types of data classification objects (entities, attributes, relationship, …)
- The model in diagram is call ERD (Entity-Relationship Diagram)
- Typical online tool for this diagram design: www.draw.io

DB Design: Identify Entities

- Entity represent objects from the real world
  - Most often they are nouns in the specification
  - For example:

> We need to develop a system that stores information about students, which are trained in various courses. The courses are held in different towns. When registering a new student the following information is entered: name, faculty number, photo and date.

  - Entities: **Student**, **Course**, **Town**

DB Design: Identify attributes

- Attributes are clarifications for the entities in the text of the specification, for example:

> We need to develop a system that stores information about students, which are trained in various courses. The courses are held in different towns. When registering a new student the following information is entered: name, faculty number, photo and date.

- Students (Entity) have the following characteristics (attributes):
  - Name, faculty number, photo, date of enlistment and a list of courses they visit

DB Design: Identify Entity Relationships

- Relationships are dependencies between the entities:

> We need to develop a system that stores information about students, which are trained in various courses. The courses are held in different towns. When registering a new student the following information is entered: name, faculty number, photo and date.

  - "Students are trained in courses" ➔ many-to-many relationship
  - "Courses are held in towns" ➔ many-to-one (or many-to-many) relationship

UNIVERSITY of GREENWICH
Alliance with FPT Education

Key

- A key is an attribute in ER diagrams which values are distinct for each individual entity in an entity set
- Primary key (PK): is the attribute or set of attributes that can uniquely identify each entity in an entity set
- Foreign key (FK): is an attribute or set of attributes that must have values from Primary Key from another entity set whenever there is some relationship between two entity sets

ERD example

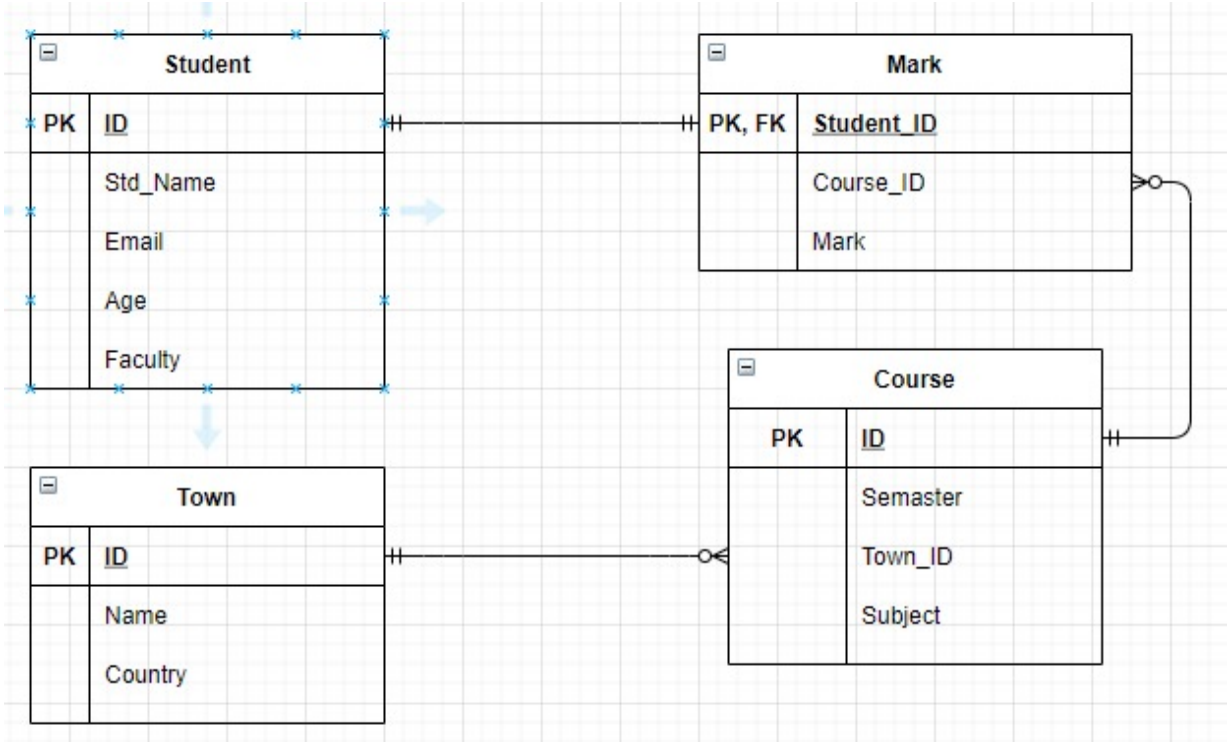

Student
PK
ID
Std_Name
Email
Age
Faculty
Mark
PK, FK
Student_ID
Course_ID
Mark
Course
PK
ID
Semaster
Town_ID
Subject
Town
PK
ID
Name
Country

TABLE RELATIONS

Relational Model in Action

Relation model

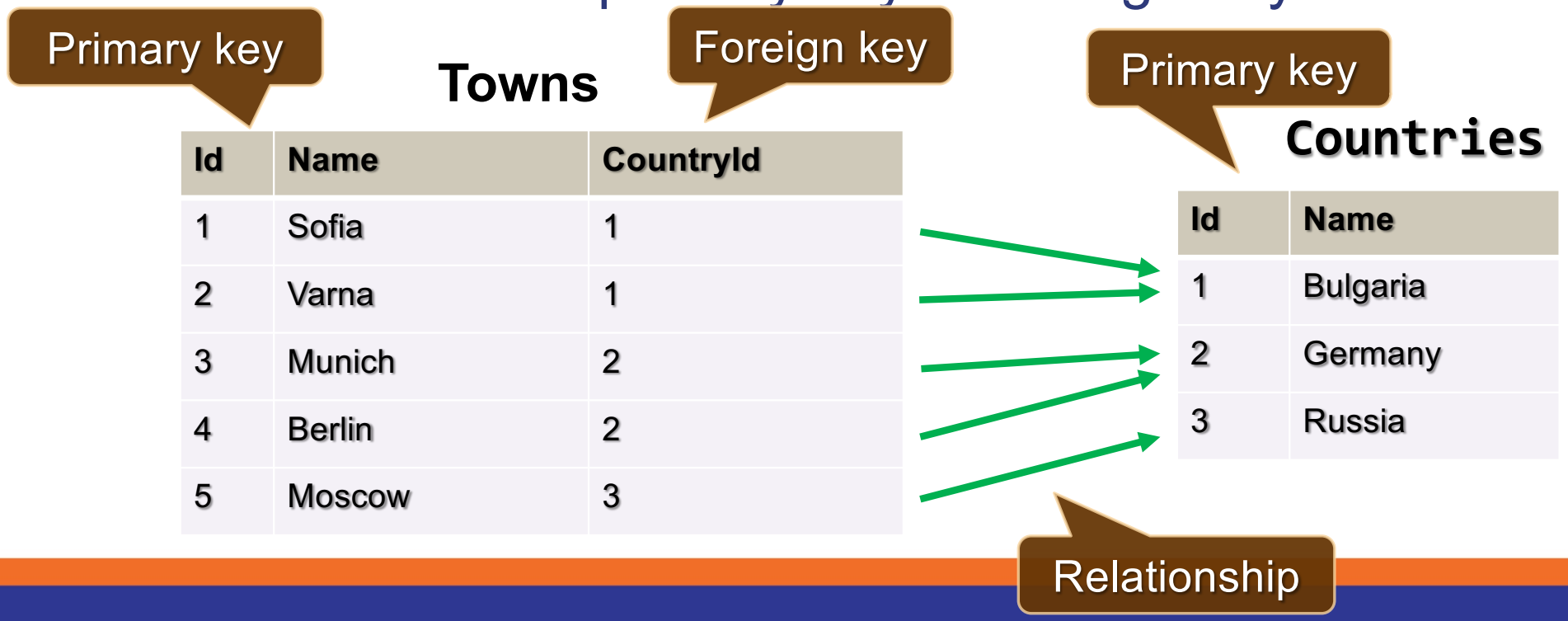
- This step will translate conceptual data model (such as ERD) into a logical data model which is suitable for implementation using the target database management system (DBMS)
- The logical data model in this course is table relation that will be implemented in a RDBMS

Table transformation

- Each entity in an ERD is mapped to a single relation table
- An attribute of an Entity in ERD is mapped to a column in a associated relation table.
- A row represents all pairings of attribute values of a associated with entity occurrences in Entity.

Table Relations

- Relationships between tables are based on interconnections: primary key → foreign key



Id	Name	CountryId
1	Sofia	1
2	Varna	1
3	Munich	2
4	Berlin	2
5	Moscow	3

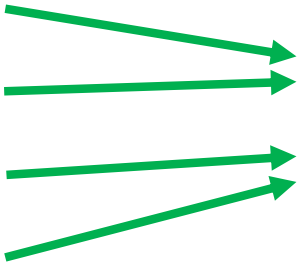
Id	Name
1	Bulgaria
2	Germany
3	Russia

Table Relations: Foreign Key

- The foreign key is an identifier of a record located in another table (usually a primary key)
- Using relationships, we refer to data instead of repeating data
  - Country name is not repeated, it is referred to by its primary key

Towns

Id	Name	CountryId
1	Sofia	1
2	Varna	1
3	Munich	2
4	Berlin	2



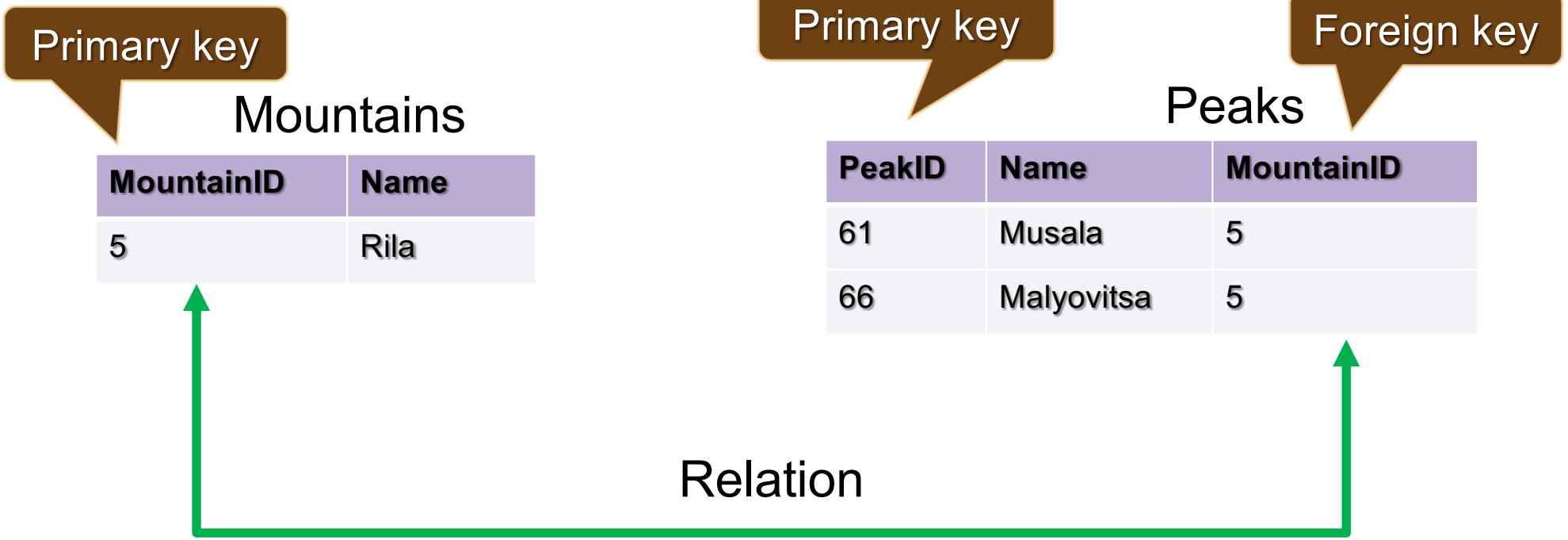
Countries

Id	Name
1	Bulgaria
2	Germany

Table Relations: Multiplicity

- **One-to-many** – e.g. country / towns
  - One country has many towns
- **Many-to-many** – e.g. student / course
  - One student has many courses
  - One course has many students
- **One-to-one** – e.g. example driver / car
  - One driver has only one car
  - Rarely used

One-to-Many/Many-to-One



One-to-Many: Tables

```
CREATE TABLE Mountains(
  MountainID INT PRIMARY KEY,
  MountainName VARCHAR(50)
)

CREATE TABLE Peaks(
  PeakId INT PRIMARY KEY,
  MountainID INT,
  CONSTRAINT FK_Peaks_Mountains
  FOREIGN KEY (MountainID)
  REFERENCES Mountains(MountainID)
)
```

Primary key

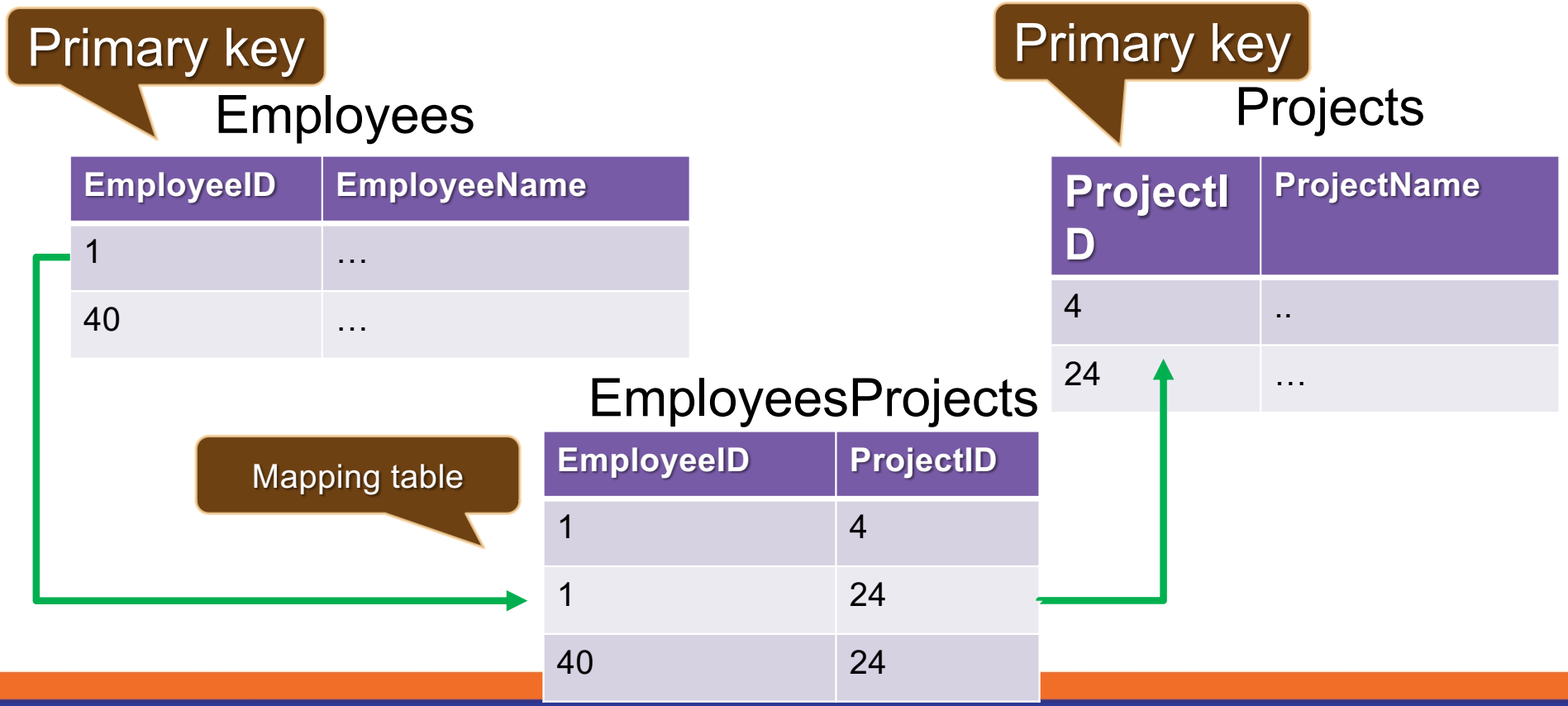
Foreign Key

One-to-Many: Foreign Key

- The table holding the foreign key is the child table
- The table holding the referenced primary key is the parent/referenced table

Many-to-Many

- Many-to-many relations use a mapping/join table



Many-to-Many: Tables

```
CREATE TABLE Employees(
  EmployeeID INT PRIMARY KEY,
  EmployeeName VARCHAR(50)
)

CREATE TABLE Projects(
  ProjectID INT PRIMARY KEY,
  ProjectName VARCHAR(50)
)
```

Many-to-Many: Mapping Table

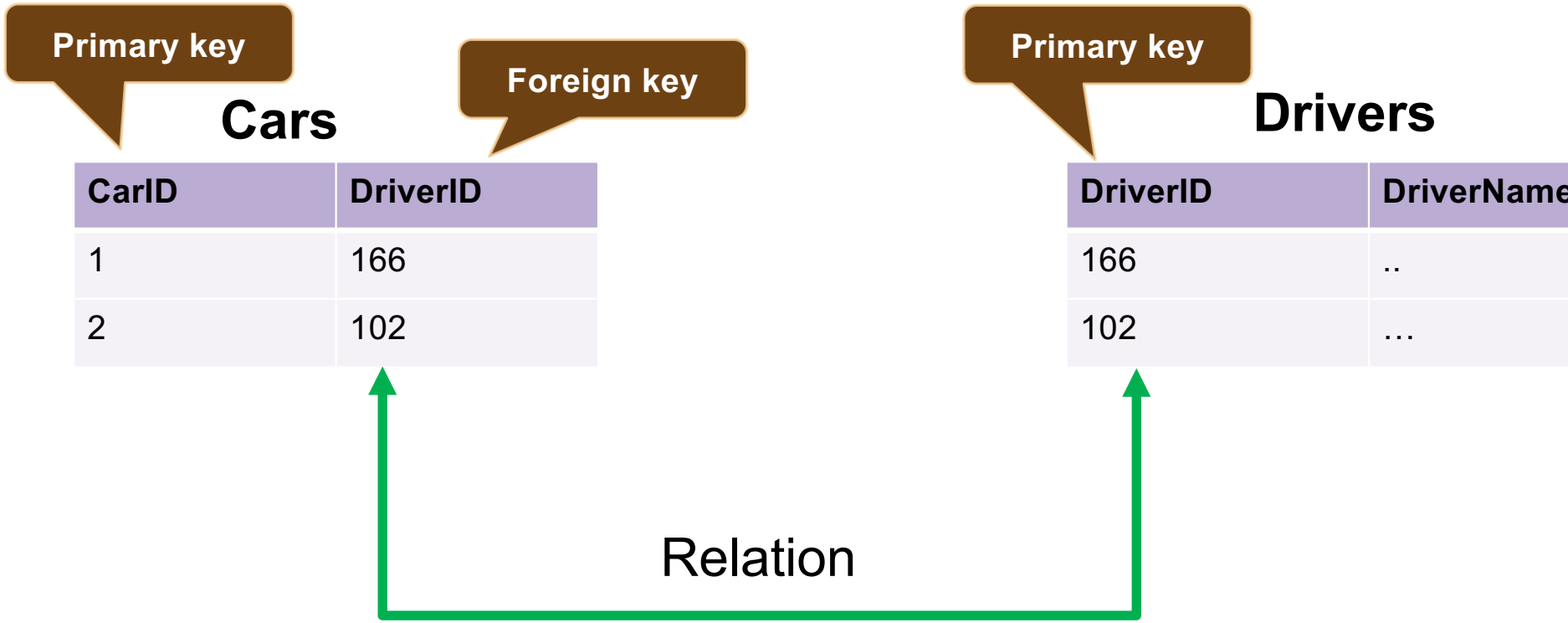
```
CREATE TABLE EmployeesProjects(
  EmployeeID INT,
  ProjectID INT,
  CONSTRAINT PK_EmployeesProjects
  PRIMARY KEY(EmployeeID, ProjectID),
  CONSTRAINT FK_EmployeesProjects_Employees
  FOREIGN KEY(EmployeeID)
  REFERENCES Employees(EmployeeID),
  CONSTRAINT FK_EmployeesProjects_Projects
  FOREIGN KEY(ProjectID)
  REFERENCES Projects(ProjectID)
)
```

Composite Primary Key

Foreign Key to Employees

Foreign Key to Projects

One-to-One


Primary key
Foreign key
Primary key
Cars
Drivers
CarID
DriverID
1
166
2
102
DriverID
DriverName
166
..
102
…
Relation

One-to-One

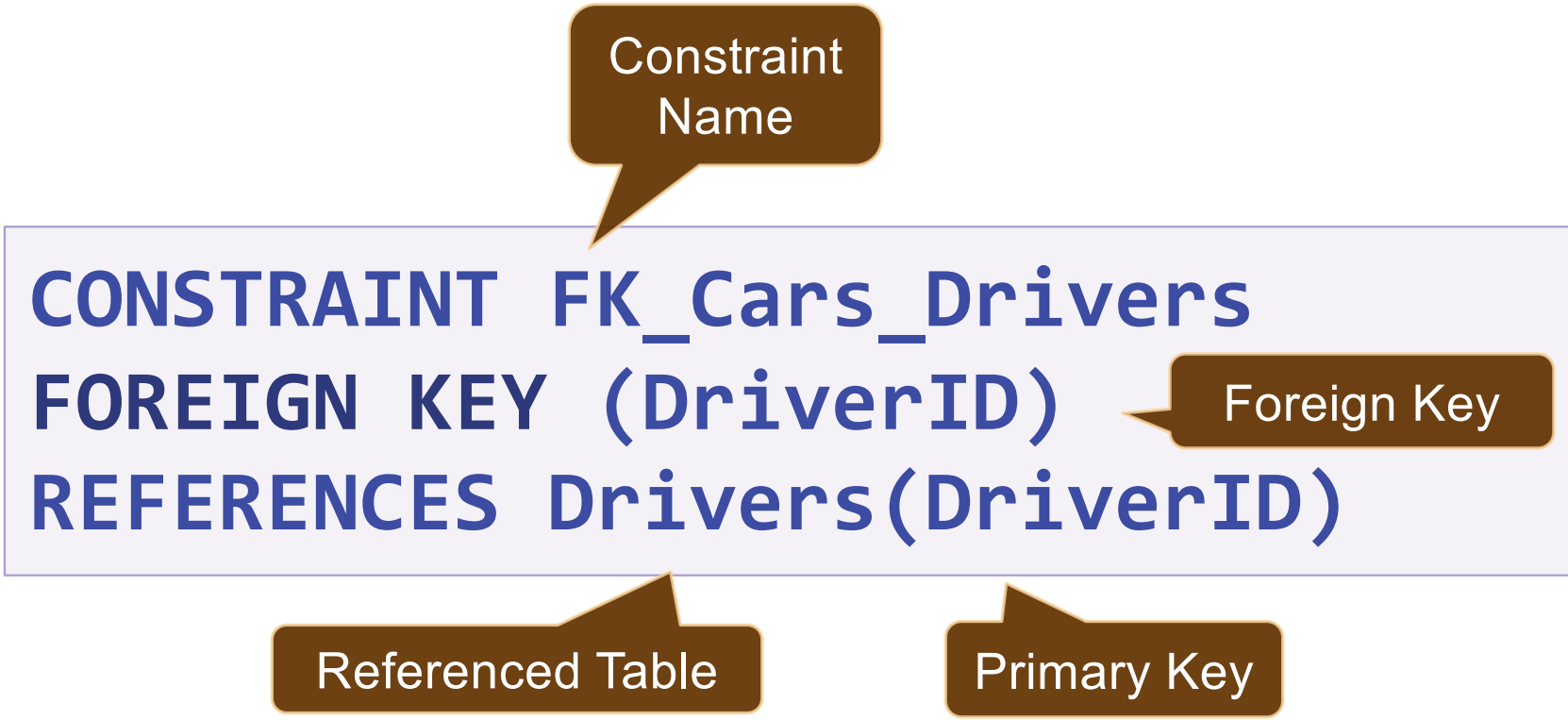
```
CREATE TABLE Drivers(
  DriverID INT PRIMARY KEY,
  DriverName VARCHAR(50)
)

CREATE TABLE Cars(
  CarID INT PRIMARY KEY,
  DriverID INT UNIQUE,
  CONSTRAINT FK_Cars_Drivers FOREIGN KEY
  (DriverID) REFERENCES Drivers(DriverID)
)
```

Primary key

One driver per car

Foreign Key

One-to-One: Foreign Key



UNIVERSITY of GREENWICH

Alliance with FPT Education

REVIEW E/R DIAGRAMS

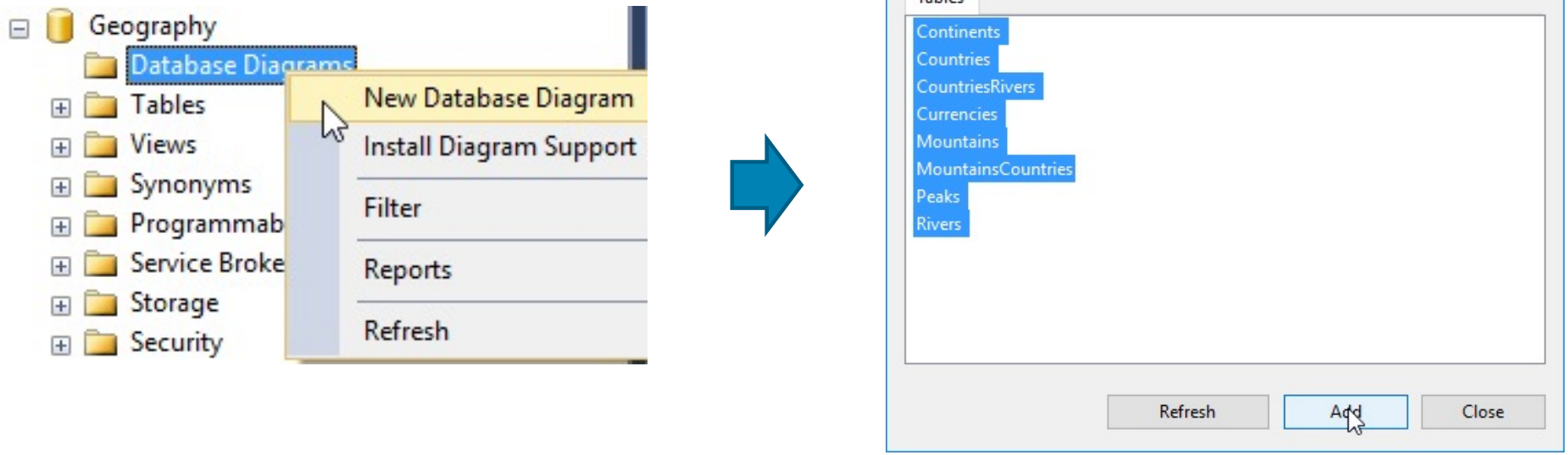
Entity / Relationship Diagrams

Relational Schema

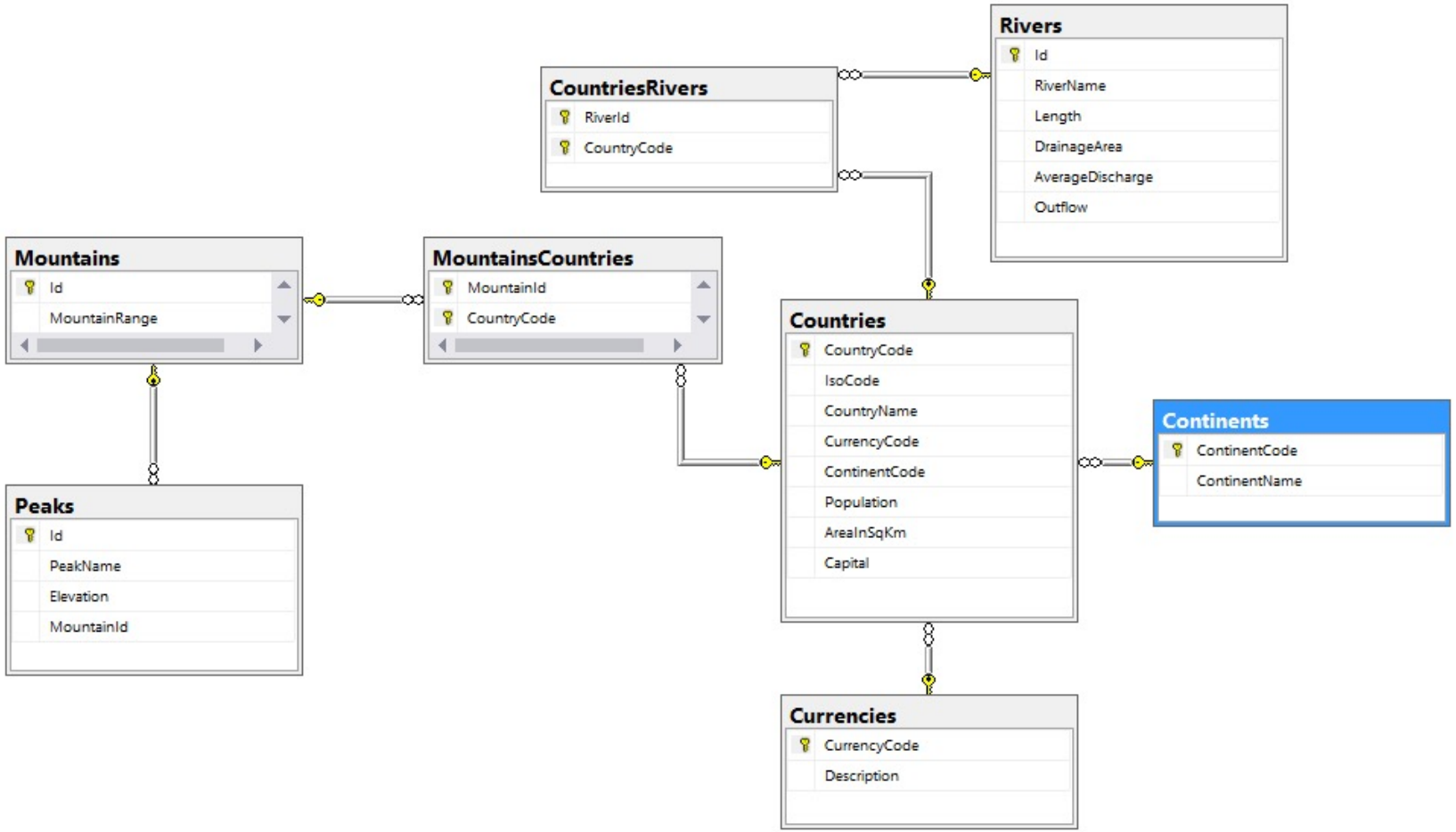
- Relational schema of a DB is the collection of:
  - The schemas of all tables
  - Relationships between the tables
  - Any other database objects (e.g. constraints)
- The relational schema describes the structure of the database
  - Doesn't contain data, but metadata
- Relational schemas are graphically displayed in Entity / Relationship diagrams (E/R Diagrams)

SSMS E/R Diagram: Usage

- Expand a database in Object Explorer
  - Right click "Database Diagrams" then select "New Database Diagram"



SSMS E/R Diagram



Summary

1. How to design multiple tables with related data?
2. What are the types of table relations?
3. How can we visualize all of our relations in a database?