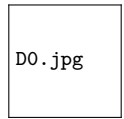
Lecture 09: ER-to-Relational Mapping

Translating Conceptual Models to Physical Schemas

Z2004: Database Management Systems



IIT Madras Zanzibar

Monday, 16 March 2026

Recap: Lecture 08

The Rules of Integrity

Last week, we learned how to mathematically protect our data from corruption using constraints:

Domain Constraints: Restricting valid inputs e.g., "`CHECK (Age >= 18)`.

Key Constraints: Enforcing uniqueness via `PRIMARY KEY` and `UNIQUE`.

Entity Integrity: A Primary Key can *never* be `NULL`.

Relationships & Logic

We also explored how tables interact and how SQL handles missing information:

- **Referential Integrity:** Foreign Keys must match an existing PK or be `NULL`.
- **Cascading Rules:** Using `ON DELETE CASCADE` or `SET NULL` to handle deleted records.
- **Three-Valued Logic:** `NULL` means unknown. Comparisons with `NULL` evaluate to `UNKNOWN`, which are excluded by `WHERE`.

Lecture Outcomes

Bridging the Gap

We have mastered the **Conceptual Design** (drawing ER diagrams) and the **Logical Rules** (Constraints).

Today, we combine them. We will learn the exact, mechanical algorithm to translate your white-board drawings directly into SQL-ready database tables.

ER Model $\Longrightarrow$ Relational Schema

By the end of today, you will...

- ✔ **Transform Entities:** Know exactly when to create a new table for Strong and Weak entities.
- ✔ **Route Foreign Keys:** Apply the rules for 1:1, 1:N, and M:N relationships to determine where the FK belongs.
- ✔ **Flatten Complexity:** Convert multivalued attributes and N-ary relationships into flat, 1NF tables.
- ✔ **Map Subclasses:** Choose the best strategy for ISA hierarchies.

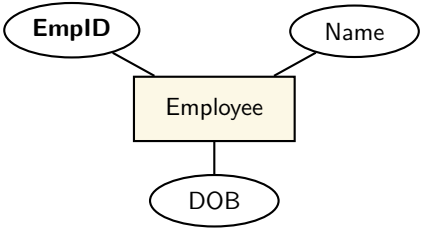
The 8 ER-to-Relational Mapping Rules

Step	ER Element	Relational Mapping Rule
1	Strong entity set	One table; PK = Entity key
2	Weak entity set	One table; Composite PK = Owner PK + Partial key
3	1:1 Binary relationship	Add FK to one side (prefer total participation side)
4	1:N Binary relationship	Add FK to the **N-side** table
5	M:N Binary relationship	New relation table with FKs to both sides
6	Multivalued attribute	New table: Entity PK + Attribute value
7	N-ary relationship ($n > 2$)	New relation table with FKs to all entities
8	ISA hierarchy	Choose one of three strategies (ER, OO, or Single Table)

★ **The Golden Rule:** After mapping, **every** ER constraint must be representable as a relational constraint (`PK`, `FK`, `UNIQUE`, `NOT NULL`, `CHECK`).

Rule 1: Mapping Strong Entities

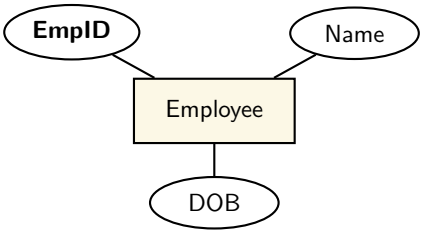
1. The ER Model



Concept: A standard entity with its own independent existence.

Rule 1: Mapping Strong Entities

1. The ER Model



Concept: A standard entity with its own independent existence.

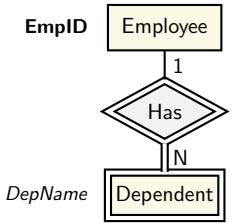
2. Relational Mapping

Rule: Create a new table. The entity's key becomes the Primary Key. All simple attributes become columns.

Employee		
EmpID (PK)	**Name**	**DOB**
101	Alice	1990-05-12
102	Bob	1995-08-22
103	Charlie	1988-11-01

Rule 2: Mapping Weak Entities

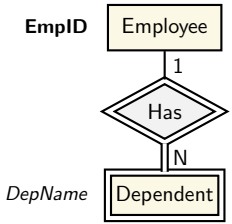
1. The ER Model



Concept: Dependent relies on Employee. *DepName* is a partial key.

Rule 2: Mapping Weak Entities

1. The ER Model



Concept: Dependent relies on Employee. *DepName* is a partial key.

2. Relational Mapping

Rule: Weak entity gets its own table. **PK** = Owner's PK + Partial Key.

Employee

EmpID (PK)	Name
101	Alice
102	Bob

Dependent

EmpID (PK/FK)	DepName (PK)	Age
101	Timmy	5
101	Sarah	8
102	Timmy	4

Rule 3: Mapping 1:1 Relationships

1. The ER Model



Concept: One Employee manages One Department.
Every Dept **must** have a manager (double line).

Rule 3: Mapping 1:1 Relationships

1. The ER Model



Concept: One Employee manages One Department. Every Dept **must** have a manager (double line).

2. Relational Mapping

Rule: Add a Foreign Key to ONE of the tables. Prefer the side with **Total Participation** to avoid NULLs.

Employee

EmpID (PK)	Name
E1	Alice
E2	Bob

Department

DeptID (PK)	DName	MgrID (FK)
D1	HR	E2
D2	IT	E1

Rule 4: Mapping 1:N Relationships

1. The ER Model



Concept: One Department has Many Employees. An Employee belongs to only One Department.

Rule 4: Mapping 1:N Relationships

1. The ER Model



Concept: One Department has Many Employees. An Employee belongs to only One Department.

2. Relational Mapping

Rule: Always place the Foreign Key on the **N-side** table.

Department (1-Side)

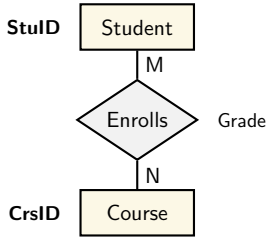
DeptID (PK)	DName
D1	Sales
D2	IT

Employee (N-Side)

EmpID (PK)	Name	DeptID (FK)
E1	Alice	D2
E2	Bob	D1
E3	Charlie	D2

Rule 5: Mapping M:N Relationships

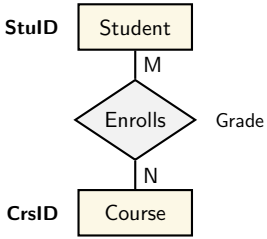
1. The ER Model



Concept: Many students take many courses. The relationship has its own attribute (Grade).

Rule 5: Mapping M:N Relationships

1. The ER Model



Concept: Many students take many courses. The relationship has its own attribute (Grade).

2. Relational Mapping

Rule: M:N requires a **Brand New Table** (Junction Table). PK = combined FKs from both entities.

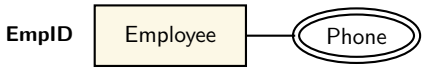
Student
StuID (PK)
S1
S2

Course
CrsID (PK)
C1
C2

Enrolls (Junction)		
StuID (PK/FK)	**CrsID (PK/FK)**	**Grade**
S1	C1	A
S1	C2	B
S2	C1	A

Rule 6: Mapping Multivalued Attributes

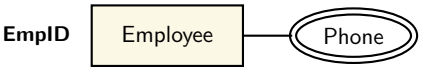
1. The ER Model



Concept: One Employee can have multiple phone numbers (Home, Cell, Work). Putting them in one cell violates 1NF (Atomicity).

Rule 6: Mapping Multivalued Attributes

1. The ER Model



Concept: One Employee can have multiple phone numbers (Home, Cell, Work). Putting them in one cell violates 1NF (Atomicity).

2. Relational Mapping

Rule: Create a **New Table**. The PK is the combination of the Entity's PK + the attribute value.

Employee

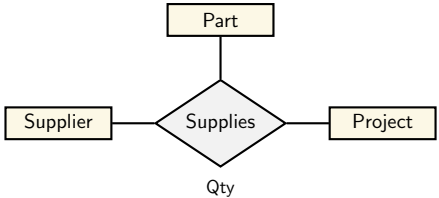
EmpID (PK)	Name
101	Alice

Emp_Phone

EmpID (PK/FK)	Phone (PK)
101	555-1234
101	555-9876

Rule 7: Mapping N-ary Relationships ($n > 2$)

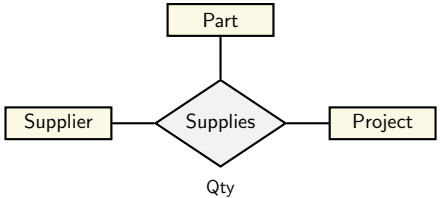
1. The ER Model (Ternary)



Concept: A specific Supplier supplies a specific Part to a specific Project.

Rule 7: Mapping N-ary Relationships ($n > 2$)

1. The ER Model (Ternary)



Concept: A specific Supplier supplies a specific Part to a specific Project.

2. Relational Mapping

Rule: Create a **New Table**. The PK is the combination of the FKs pointing to all participating entities.

(Assuming Supplier, Part, and Project tables exist)

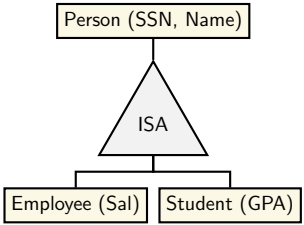
Supplies

SupID	PartID	ProjID	Qty
S1	P1	J1	500
S1	P2	J1	200
S2	P1	J2	100

Primary Key: `(SupID, PartID, ProjID)`
Foreign Keys:
`SupID` → `Supplier(SupID)`
`PartID` → `Part(PartID)`
`ProjID` → `Project(ProjID)`

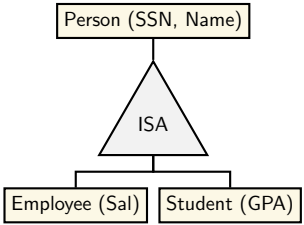
Rule 8: Mapping ISA Hierarchies (Subtyping)

1. ER Model



Rule 8: Mapping ISA Hierarchies (Subtyping)

1. ER Model



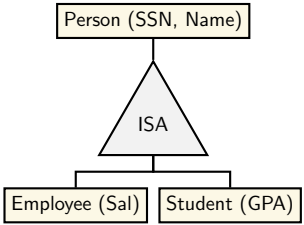
2. Three Mapping Strategies

Strategy A: The ER Approach (3 Tables)

- `Person(SSN, Name)`
- `Employee(SSN_FK, Sal)`
- `Student(SSN_FK, GPA)`

Rule 8: Mapping ISA Hierarchies (Subtyping)

1. ER Model



2. Three Mapping Strategies

Strategy A: The ER Approach (3 Tables)

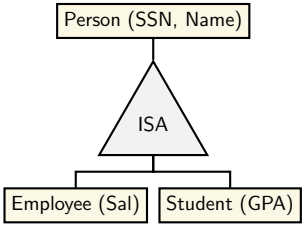
- `Person(SSN, Name)`
- `Employee(SSN_FK, Sal)`
- `Student(SSN_FK, GPA)`

Strategy B: The Object-Oriented Approach (2 Tables)

- `Employee(SSN, Name, Sal)`
- `Student(SSN, Name, GPA)`
- *(Only works if Subclasses are Total/Disjoint)*

Rule 8: Mapping ISA Hierarchies (Subtyping)

1. ER Model



2. Three Mapping Strategies

Strategy A: The ER Approach (3 Tables)

- `Person(SSN, Name)`
- `Employee(SSN_FK, Sal)`
- `Student(SSN_FK, GPA)`

Strategy B: The Object-Oriented Approach (2 Tables)

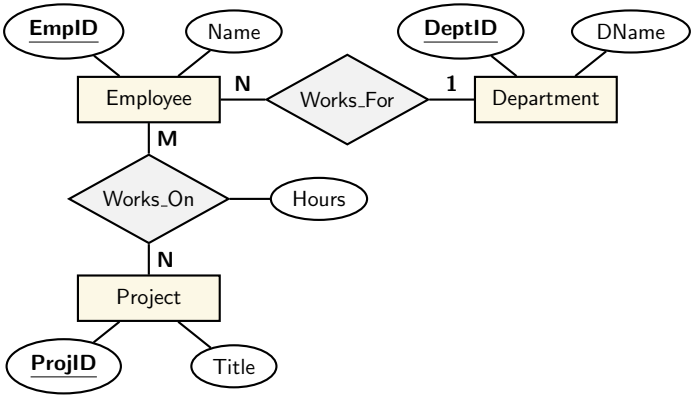
- `Employee(SSN, Name, Sal)`
- `Student(SSN, Name, GPA)`
- *(Only works if Subclasses are Total/Disjoint)*

Strategy C: The Single Table Approach (1 Table)

- `Person(SSN, Name, Type, Sal, GPA)`
- *(`Type` flags if they are Emp or Stu. Lots of NULLs!)*

Putting It All Together: A Complete ER Diagram

The Conceptual Model (Mini-Company)



Challenge for the class: How many tables will this ER diagram produce? Which tables get the Foreign Keys?

The Resulting Relational Schema

1. Strong Entities (Rule 1)

Department

DeptID (PK)	DName
D1	HR
D2	IT

Project

ProjID (PK)	Title
P99	Server Migration

2. 1:N Relationship (Rule 4)

(FK goes to the N-side: Employee)

Employee

EmpID (PK)	Name	DeptID (FK)
E1	Alice	D2
E2	Bob	D1

3. M:N Relationship (Rule 5)

(Creates a new junction table)

Works_On

EmpID (PK/FK)	ProjID (PK/FK)	Hours
E1	P99	40
E2	P99	15

Quiz: ER → Relational Mapping (Modern Example)

Scenario — Campus Food Delivery App:

- **Customer**(CustID, Name, Email) places **Orders**(OrderID, OrderTime).
- Each **Order** contains **many MenuItems**, and a **MenuItem** can appear in many Orders.
- **Restaurant**(RestID, Name) offers MenuItems (each MenuItem belongs to exactly one Restaurant).
- For each MenuItem in an Order, we must store **Quantity**.

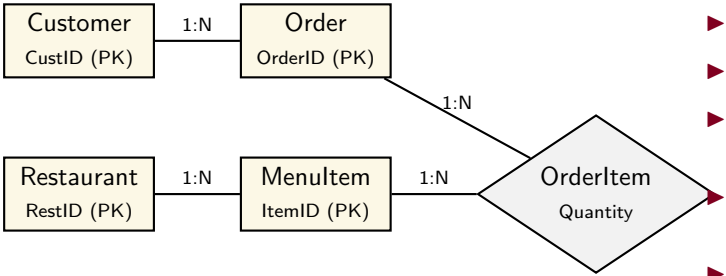
Your task (10 minutes):

1. Draw the ER diagram (entities, relationships, cardinalities).
2. Convert it to a relational schema with **PKs** and **FKs**.

Hint: The M:N between Order and MenuItem needs an extra table to store Quantity.

Solution: ER Diagram + Relational Schema

ER (one valid answer)



We resolve the M:N (Order–MenuItem) using associative entity/table **OrderItem**.

Relational schema

- **Customer**(<u>CustID</u>, Name, Email)
- **Restaurant**(<u>RestID</u>, Name)
- **MenuItem**(<u>ItemID</u>, ItemName, Price, RestID (FK → Restaurant))
- **Orders**(<u>OrderID</u>, OrderTime, CustID (FK → Customer))
- **OrderItem**(<u>OrderID</u>, <u>ItemID</u>, Quantity, OrderID (FK → Orders), ItemID (FK → MenuItem))

Composite PK on OrderItem prevents duplicate items per order (common choice).

Example Instances (1/2): Restaurant & MenuItem

Restaurant

RestID (PK)	Name
R01	Zanzibar Bites
R02	Night Owl Cafe
R03	Harbor Grill
R04	Spice Lab
R05	Green Bowl

Used by: MenuItem.RestID (FK)

MenuItem

ItemID (PK)	ItemName	Price	RestID (FK)
I10	Chips Masala	3.0	R01
I11	Shawarma Wrap	4.5	R01
I20	Iced Coffee	2.5	R02
I21	Veg Samosa	1.5	R02
I30	Grilled Chicken	6.0	R03
I31	Paneer Bowl	5.0	R05

Example Instances (2/2): Customer, Orders & OrderItem

Customer

CustID (PK)	Name	Email
C01	Asha	asha@campus.edu
C02	Omar	omar@campus.edu
C03	Neema	neema@campus.edu
C04	Yara	yara@campus.edu
C05	Raj	raj@campus.edu

Orders

OrderID (PK)	OrderTime	CustID (FK)
O1001	09:10	C01
O1002	09:35	C02
O1003	10:05	C03
O1004	12:15	C04
O1005	18:40	C05

OrderItem (Junction Table)

OrderID (PK/FK)	ItemID (PK/FK)	Quantity
O1001	I10	2
O1001	I20	1
O1002	I11	1
O1003	I20	2
O1004	I31	1
O1005	I30	1

FKs: Orders.CustID $\rightarrow$ Customer, OrderItem.OrderID $\rightarrow$ Orders, OrderItem.ItemID $\rightarrow$ MenuItem.

Lecture Summary

Key Takeaways

- **Entities → Tables:** Strong entities become base tables. Weak entities pull in their owner's PK.
- **1:N Relationships:** The Foreign Key **always** goes on the "Many" (N) side.
- **M:N Relationships:** Must be broken down into a brand-new Junction Table.
- **Multi-valued Attributes:** Get spun off into their own separate tables to preserve 1NF.

Next Lecture (L10)

Relational Algebra: The Math of Queries

- We have built the database. Now, how do we ask it questions?
- Selection (σ) and Projection (π).
- Combining data: The Mathematics of Joins ($\bowtie$).

Reading Assignment

Read **Silberschatz, Chapter 6** (Formal Relational Query Languages) before Lecture 10.

Course Grading

Assessment Breakdown

Component	Weight
Quiz 1	20%
Midsem Exam	20%
Project	20%
Endsem Exam	30%
Assignments (ICA)	10%

Assignments (ICA)

- Total assignments: **5**
- Assignment 1: **Tuesday, 17 March 2026**, **9:00–9:50 AM**