Today

- ER -> Relational Review
- Merge Rule
- SQL Examples

Mapping Entity Sets

- Create table for each entity set
- Map attributes to fields
- Declare primary key

Mapping Relationship Set (no key constraints)

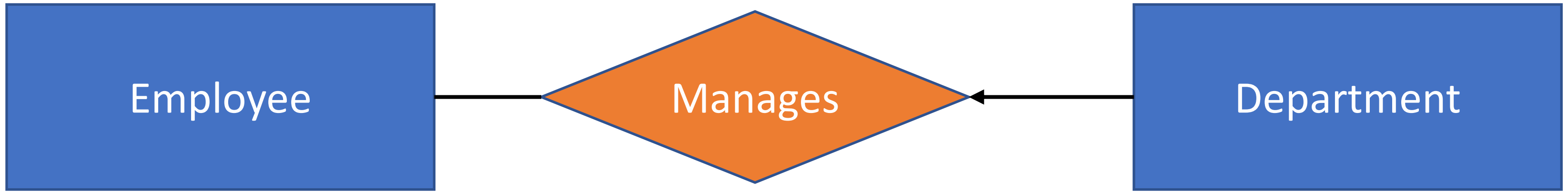
- Create table for relationship set
- Add primary keys of entity sets participating in the relationship as primary keys of the relation
- Map attributes of the relationship to fields

Mapping Relationship Set (with key constraints)

- Option 1:
  - Create table for relationship set
  - Add primary keys of participating entity sets as fields
  - Map attributes of the relationship to fields
  - Declare primary key using key fields from source entity set (where the arrow is coming from)

Mapping Relationship Set (with key constraints)

- Option 1:



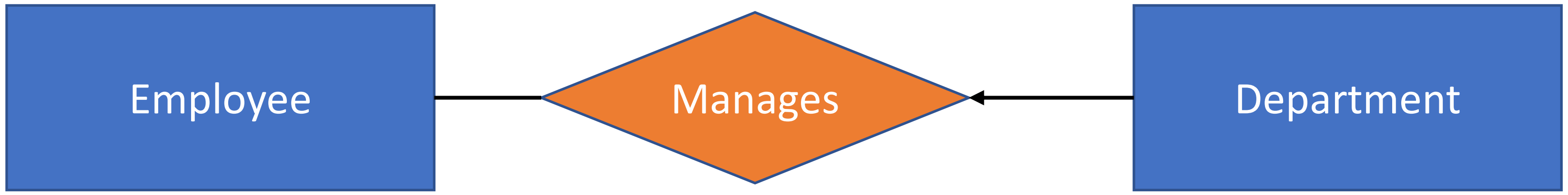
```
CREATE TABLE manages(
     employee_id           integer,
     department_id         integer,
     PRIMARY KEY (department_id),
     FOREIGN KEY (employee_id) REFERENCES Employees,
     FOREIGN KEY (department_id) REFERENCES Departments);
```

Mapping Relationship Set (with key constraints)

- Option 2
    - Add primary key of target entity as a field in the source

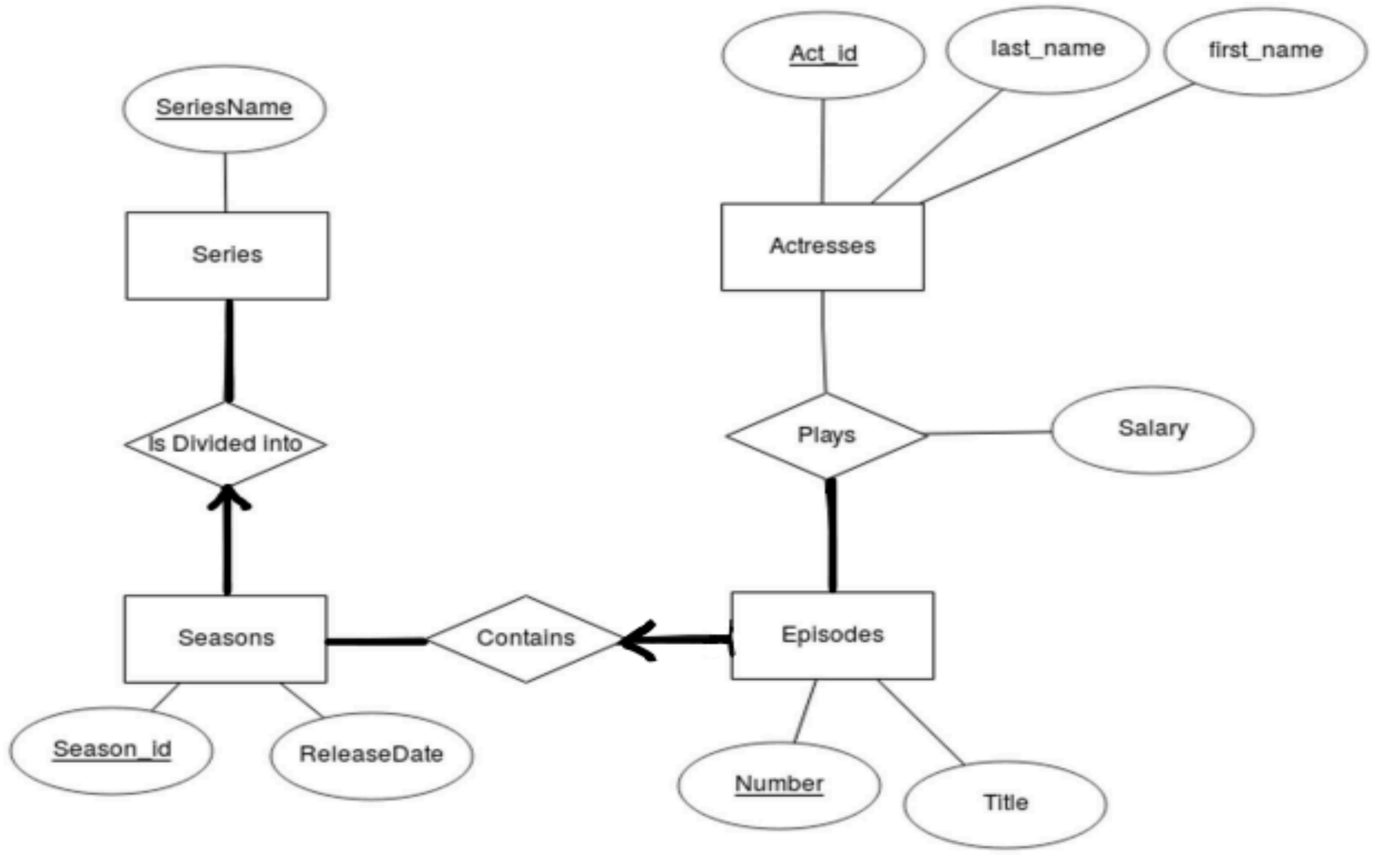
Mapping Relationship Set (with key constraints)

- Option 2:

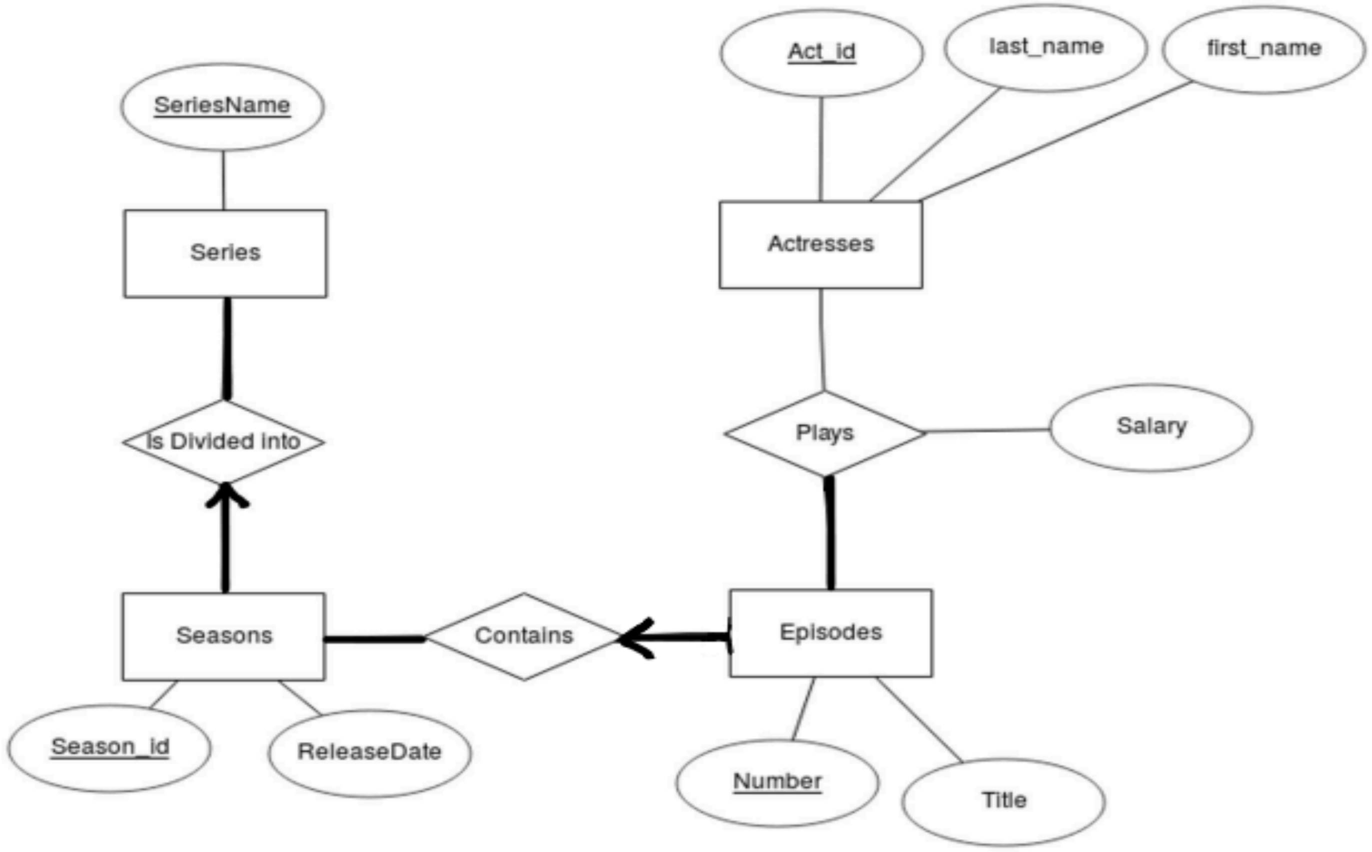


```
CREATE TABLE department(
      department_id             integer,
      department_name           varchar(20),
      employee_id               integer,
      PRIMARY KEY (department_id),
      FOREIGN KEY (employee_id) REFERENCES Employees);
```

- Note: if we declare employee_id as NOT_NULL, can enforce participation constraint. Cannot easily do this with Option 1.



Series(<u>SeriesName:</u>char(50))
Seasons(<u>Season_Id:int</u>, ReleaseDate:datetime, #SeriesName:int)
Episode(<u>Number:int</u>, Title:char(50), #Season_id:int)
Actresses(<u>Act_id</u>:int, last_name:char(50), first_name:char(50))
Plays(<u>#Number:int, #Act_id:int</u>, salary:int)



Series(<u>SeriesName</u>:char(50))
Seasons(<u>Season_id</u>:int, ReleaseDate:datetime)
Episodes(<u>Number</u>:int, Title:char(50))
Actresses(<u>Act_id</u>:int, last_name:char(50), first_name:char(50))

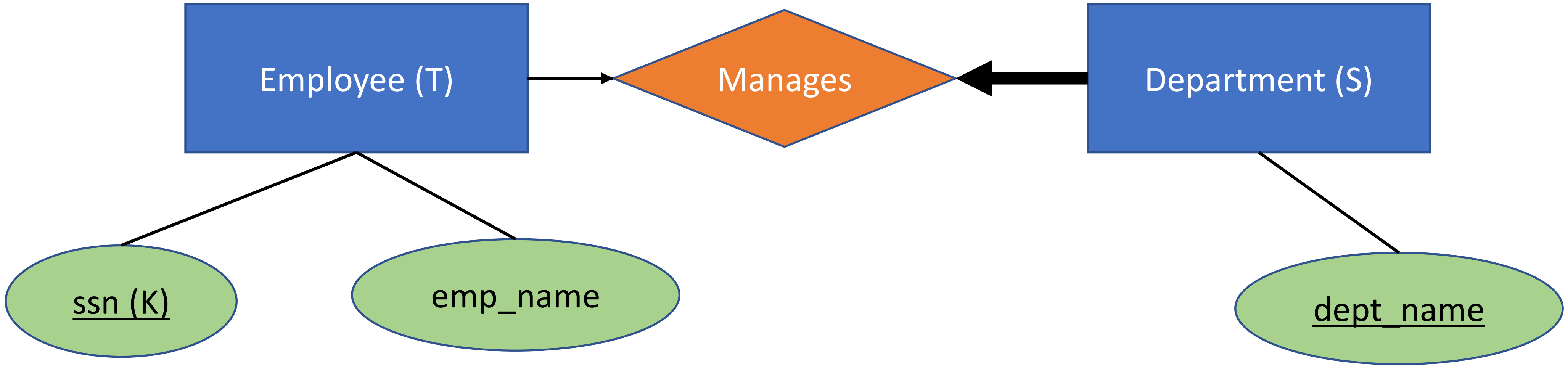
IsDividedInto(#SeriesName:char(50), #<u>Season_id</u>:int)
Contains(#Season_id:int, #<u>Number</u>:int)
Plays(#<u>Act_id</u>:int, #<u>Number</u>:int, Salary:int)

Merge Rule

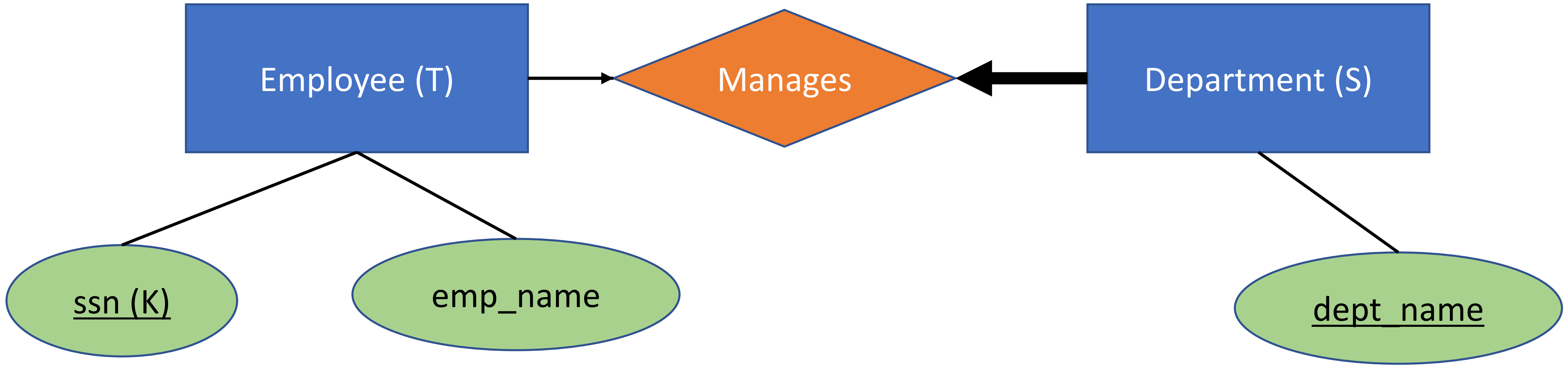
- DB Table Design Principles
  1. Repetition of data is bad if unnecessary
     1. Wastes space
     2. Need to verify and maintain consistency
  2. Having fewer tables is better because queries require fewer joins:
     a. Easier to write queries
     b. Less expensive to execute
  3. Having table with too many nulls is undesirable

Merge Rule

- If you have two tables of the form T(K, X) and S(K2, Y), Y *NOT NULL*, where K2 is a foreign key referencing T(K), then you can merge S into T to get instead a single table TS(K, X, Y). Column Y will have NULLs for all keys in T but not S.



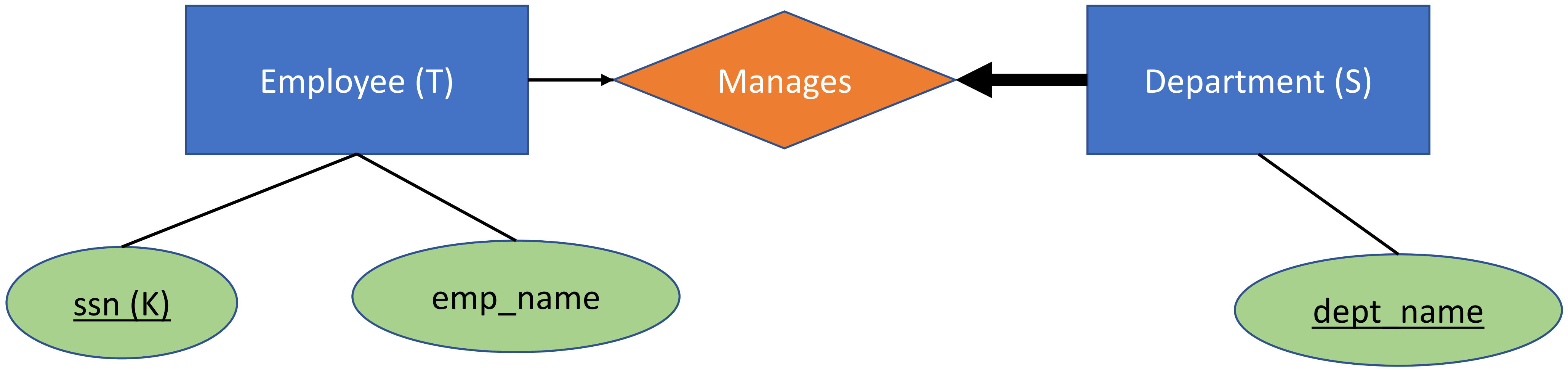
Merge Rule



ssn	emp_name
1	alice
2	bob
3	charlie

dept_name	ssn
finance	1
sales	3

Merge Rule



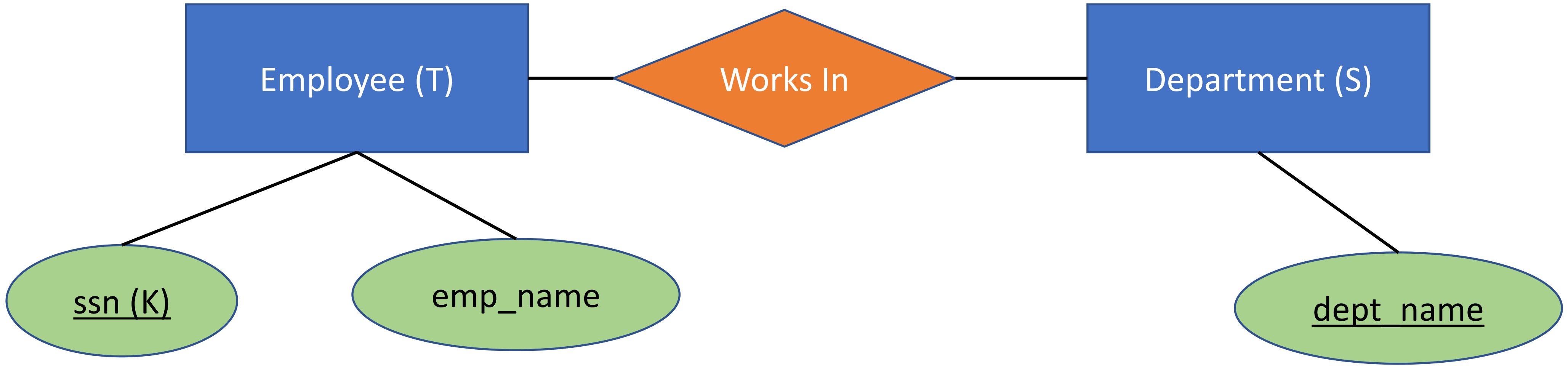
ssn	emp_name
1	alice
2	bob
3	charlie

Merged table

ssn	emp_name	dept_name
1	alice	finance
2	bob	NULL
3	charlie	sales

dept_name	ssn
finance	1
sales	3

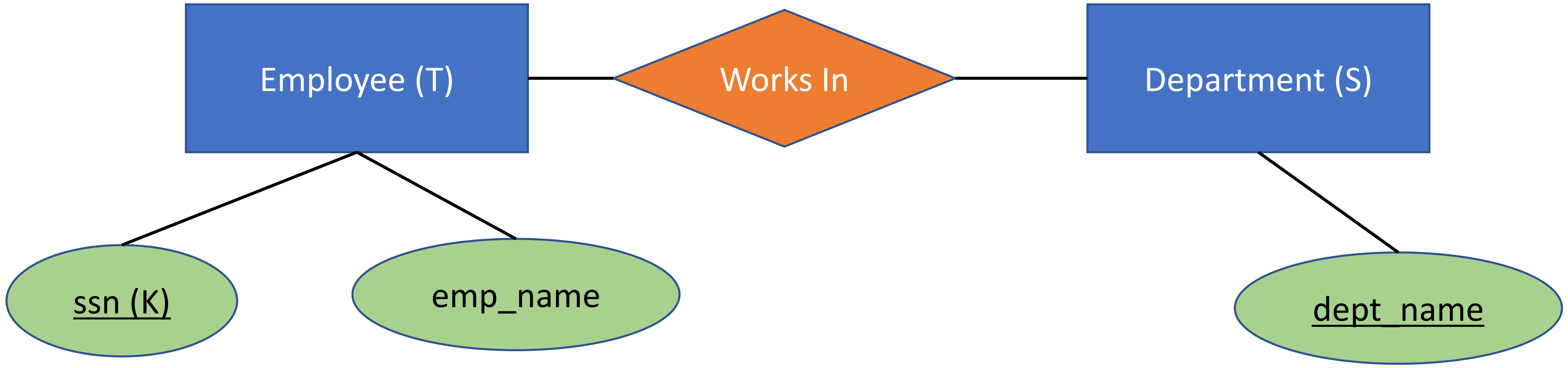
Merge Rule



ssn	emp_name
1	alice
2	bob
3	charlie

dept_name	ssn
finance	1
legal	1
sales	3

Merge Rule



ssn	emp_name
1	alice
2	bob
3	charlie

ssn	emp_name	dept_name
1	alice	finance
1	alice	legal
3	bob	NULL
4	charlie	sales

dept_name	ssn
finance	1
legal	1
sales	3

SQL Examples