2. Relational Model Exercises
E-R Diagram for a Banking Enterprise
Tables for ER diagram

- **Entities**
  - Branch (branch-name, branch-city, assets)
  - Customer (customer-id, customer-name, customer-street, customer-city)
  - Loan (loan-number, amount)
  - Employee (employee-id, employee-name, telephone-number, start-date)
  - Account (account-number, balance)
  - Savings-account (account-number, interest-rate)
  - Checking-account (account-number, overdraft-amount)
Tables for ER diagram

• Weak Entity
  – Payment (loan-number, payment-number, payment-date, payment-amount)

• Multi-valued attribute
  – Dependent (employee-id, dependent-name)
Tables for ER diagram

- **Many-to-many relationships**
  - Borrower (customer-id, loan-number)
  - Depositor (customer-id, account-number, access-date)

- **One-to-many relationships**
  - Loan-branch is represented in Loan (loan-number, amount, branch-name)
  - Cust-banker is represented in Customer (customer-id, customer-name, customer-street, customer-city, employee-id, type)
  - Works-for is represented in Employee (employee-id, employee-name, telephone-number, start-date, manager-id)
Convert the ER diagram into tables.
Tables for ER diagram

- **Entities**
  - Route (number, departure, destination)
  - Driver (id, name, phone)
  - Bus (license, capacity)

- **Weak Entity**
  - Schedule (number, departure-time)

- **Relationships**
  - Drives (number, departure-time, id)
  - Bus-in-use (license, number, departure-time)
We want to design a relational database schema to represent information about an internal training program of a large company. Design an ER diagram based on the description given below:

- The database keeps information about every employee in the company. Each employee has a name and a unique employee number.
- Each course of the training program has a unique course number and a name.
- The courses are taught and taken by employees of the company.
- A course can be offered many times. Each offering has an offering number, which is unique within each course, a day and time.
- An offering is taught by exactly one employee.
- The database stores the grades of employees who took courses.
Solution

Employee (EmpNo, Name)
Course (CourseNo, Name)
Offering (CourseNo, OfferingNo, Day, Time, EmpNo)
Enrolled (CourseNo, OfferingNo, EmpNo, Grade)
Let the following relational schema:

\[ B(X, W), \]
\[ C(P, Q, X) \] where \( X \) is defined as NOT NULL,
\[ D(P, X, R) \]

Give an ER diagram for the above relational schema.
Solution

Diagram:

- B
- \( \text{any\_name} \)
- C
- X
- W
- D
- P
- Q
- R
Find ids of sailors who’ve reserved boat with bid 103

- We use the following database schema

  **Sailors** (sid, sname),
  **Reserves** (sid, bid, date),
  **Boats** (bid, bname, color)

Question: Give an ER diagram for the above tables.

- $\pi_{\text{sid}}(\sigma_{\text{bid}=103}\text{Reserves})$
Find names of sailors who’ve reserved boat with bid 103

Solution 1:

\[ \pi_{\text{sname}}(\sigma_{\text{Reserves.sid}=\text{Sailors.sid} \land \text{bid}=103}(\text{Reserves} \times \text{Sailors})) \]

Solution 2:

\[ \pi_{\text{sname}}(\sigma_{\text{Reserves.sid}=\text{Sailors.sid}}((\sigma_{\text{bid}=103}\text{Reserves}) \times \text{Sailors})) \]

Solution 3:

\[ \pi_{\text{sname}}((\sigma_{\text{bid}=103}\text{Reserves}) \JOIN_{\text{sid}}\text{Sailors}) \]
Find names of sailors who’ve reserved a red boat

\[ \text{Sailors}(\text{sid}, \text{sname}), \text{Reserves}(\text{sid}, \text{bid}, \text{date}), \text{Boats}(\text{bid}, \text{bname}, \text{color}) \]

Solution 1:

\[ \pi_{\text{sname}}((\sigma_{\text{color}=\text{red}} \text{Boats}) \text{ JOIN}_{\text{bid}} \text{Reserves} \text{ JOIN}_{\text{sid}} \text{Sailors}) \]

Can you give a more efficient solution?

\[ \pi_{\text{sname}}((\pi_{\text{bid}}(\sigma_{\text{color}=\text{red}} \text{Boats})) \text{ JOIN}_{\text{bid}} \text{Reserves} \text{ JOIN}_{\text{sid}} \text{Sailors}) \]

query optimization: real systems do such optimizations based on algebra
Find the names of sailors who’ve reserved a red or a green boat

Sailors(sid, sname), Reserves(sid, bid, date), Boats(bid, bname, color)

• Can identify all red or green boats, then find sailors who’ve reserved one of these boats:

\[ \pi_{sname}((\sigma_{\text{color}=\text{red}} \lor \text{color}=\text{green} \text{ Boats}) \Join_{\text{bid}} \text{Reserves} \Join_{\text{sid}} \text{Sailors}) ]

What happens if \( \lor \) is replaced by \( \land \) in this query?
Find the names of sailors who’ve reserved a red and a green boat

\[ \text{Sailors}(\text{sid}, \text{sname}), \text{Reserves}(\text{sid, bid, date}), \text{Boats}(\text{bid, bname, color}) \]

- **Previous approach won’t work!** Must identify sailors who’ve reserved red boats, sailors who’ve reserved green boats, then find the intersection:

  Is this solution correct?

  \[
  \pi_{\text{sname}}((\sigma_{\text{color}=\text{red}} \text{Boats}) \text{ JOIN}_{\text{bid}} \text{Reserves JOIN}_{\text{sid}} \text{Sailors}) \cap \pi_{\text{sname}}((\sigma_{\text{color}=\text{green}} \text{Boats}) \text{ JOIN}_{\text{bid}} \text{Reserves JOIN}_{\text{sid}} \text{Sailors})
  \]

  Correct solution:

  \[
  \pi_{\text{sname}}
  \left[ \pi_{\text{sid, sname}}((\sigma_{\text{color}=\text{red}} \text{Boats}) \text{ JOIN}_{\text{bid}} \text{Reserves JOIN}_{\text{sid}} \text{Sailors}) \cap \pi_{\text{sid, sname}}((\sigma_{\text{color}=\text{green}} \text{Boats}) \text{ JOIN}_{\text{bid}} \text{Reserves JOIN}_{\text{sid}} \text{Sailors}) \right]
  \]
Find ids of sailors who have made at least two reservations on the same date

Sailors(sid, sname), Reserves(sid, bid, date), Boats(bid, bname, color)

We have to use rename: \( p(R1, \text{Reserves}), p(R2, \text{Reserves}) \)

\[
\pi_{R1.sid} (\sigma_{R1.sid=R2.sid \land R1.date=R2.date \land R1.bid \neq R2.bid} (R1 \times R2))
\]

Or equivalently:

\[
\pi_{R1.sid} (\sigma_{R1.sid=R2.sid \land R1.date=R2.date \land R1.bid \neq R2.bid} (p(R1, \text{Reserves}) \times p(R2, \text{Reserves})))
\]

\[
\pi_{R1.sid} (p(R1, \text{Reserves}) \ \text{JOIN} \ \pi_{R1.sid} (p(R2, \text{Reserves})) \ \text{ON} \ R1.sid=R2.sid \land R1.date=R2.date \land R1.bid \neq R2.bid)
\]

What happens if we omit \( R1.date=R2.date \)?
What happens if we omit \( R1.bid \neq R2.bid \)?
Find the ids of sailors who’ve reserved all boats

Sailors\( (\text{sid, sname}) \), Reserves\( (\text{sid, bid, date}) \), Boats\( (\text{bid, bname, color}) \)

Uses division; schemas of the input relations must be carefully chosen:

\[
(\pi_{\text{sid, bid}} \text{Reserves}) / \pi_{\text{bid}} \text{Boats}
\]

What about the query: find the ids of sailors who have reserved all red boats

\[
(\pi_{\text{sid, bid}} \text{Reserves}) / \pi_{\text{bid}} (\sigma_{\text{color=red}} \text{Boats})
\]

What about the query: find the names of sailors who have reserved all red boats

\[
\pi_{\text{sname}} (\text{Sailors JOIN}_{\text{sid}} (\pi_{\text{sid, bid}} \text{Reserves}) / \pi_{\text{bid}} (\sigma_{\text{color=red}} \text{Boats}))
\]