Schema Design and Join
Outline

- An example of reducing E/R model to relational schema
- Examples of join
E/R model
Step 1: Entities

- Entities that can be uniquely identified by its own attribute(s), e.g.
  - `customer_id` identifies customer
  - `account_num` identifies account

- Four tables
  - `branch=(branch_name, branch_city, assets)`
  - `customer=(customer_id, customer_name, customer_addr)`
  - `loan=(loan_number, amount)`
  - `account=(account_num, balance)`
Step 2: Many-to-many relationship

- Two tables: attributes are primary keys of participating entities
  - \( \text{borrower} = (\text{customer_id, loan_number}) \)
  - \( \text{depositor} = (\text{customer_id, account_number}) \)
Step 3: Many-to-one relationship

- Two options:
  - A new table for the relationship
  - Add extra attribute(s) to “many” side
- New table
  - `loan_branch(loan_number, branch_name)`
- Add an extra attribute to `loan`
  - `loan=(loan_number, amount, branch_name)`
Step 4: ISA relationship

- Two tables
  - savings_account=(account_number, interest_rate)
  - checking_account=(account_number, overdraft_amount)
Step 5: Identify primary key & foreign key
Join: connect records from different tables

- Equal-join
  - condition is “=”
  - In many cases, they are primary key-foreign key relationships

- Non-equal join
  - condition is “<” “>” or “<>”
Equal-join (primary key-foreign key)

- Find all the customers who have loan account in the branch “Gilman Dr”
  - SELECT c.customer_name
  - FROM customer c, borrow b, loan l
  - WHERE c.customer_id = b.customer_id AND b.loan_number = l.loan_number AND l.branch_name = ‘Gilman Dr’
Equal Join (ordinary attributes)

- **Schema**
  - Candidate(\textit{cid}, \textit{name}, \textit{expect\_salary}, \textit{city})
  - Job(\textit{jid}, \textit{company\_name}, \textit{salary}, \textit{city})

- \textit{Find candidate-job pairs such that they are in the same city and job’s salary is at most 60k}
  - SELECT *
  - FORM candidate c, job j
  - WHERE c.city = j.city AND j.salary <= 60
Non-equal Join

- Schema
  - Candidate(
    - cid
    - name
    - expect_salary
    - city
  )
  - Job(
    - jid
    - company_name
    - salary
    - city
  )

- Find candidate-job pairs such that candidate’s expectation salary is lower than job’s
  - SELECT *
  - FROM candidate c, job j
  - WHERE c.expect_salary < j.salary
backup
Primary key

- $\text{customer}= (\text{customer_id, customer_name, customer_addr})$
  - Two rows $t_1, t_2$ in the $\text{customer}$ table
  - If $t_1.\text{customer_id} = t_2.\text{customer_id}$.
    - then $t_1.\text{customer_name} = t_1.\text{customer_name}$ AND $t_1.\text{customer_addr} = t_2.\text{customer_addr}$
Foreign key

- Foreign key identifies a column or a set of columns in one (referencing) table that refers to a column or set of columns in another (referenced) table.
- The values in one row of the referencing columns must occur in a single row in the referenced table.
- Example
  - \( customer=(customer\_id, customer\_name, customer\_addr) \)
  - \( borrower=(customer\_id, loan\_number) \)
  - \( borrower.customer\_id \) must exist in customer