Schema Design

2/4/09

Jerry Chou
Reducing E/R Model to Relational Schema

- **E/R Model:** Entity-Relationship Model
  - An abstract and conceptual representation of data

- **Relational Schema:**
  - A database model based on predicate logic

<table>
<thead>
<tr>
<th>Bank</th>
<th>Location</th>
<th>Assets</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOA</td>
<td>NY</td>
<td>1000</td>
</tr>
<tr>
<td>Chase</td>
<td>SF</td>
<td>3000</td>
</tr>
</tbody>
</table>

- **Goal:** Construct appropriate relational schema tables to capture the relationship in E/R model.
E/R Model Example
Relational Schema

customer
- customer_id
- customer_name
- customer_addr

borrow
- customer_id
- loan_number

loan
- loan_number
- amount
- branch_name

branch
- branch_name
- branch_city
- assets

depositor
- customer_id
- account_num

account
- account_num
- account
- branch_name

checking
- account_num
- interest_rate

saving
- account_num
- overdraft_amount
E/R Model

- Entity: can be independent exist and uniquely identified. (nouns)
- Relation: connection between two entities. (verbs)
- Attribute: associate with entity or relation.
- Primary key: uniquely identifying attribute of an entity.
Step 1: Entities

- Find entities that can be uniquely identified by its own attribute(s), e.g.
  - `customer_id` identifies customer
  - `account_num` identifies account
Step1: Entities
Step 1: Entities

- Find 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)`
Step2: Relation
Step 2.1: Many-to-One Relation

- Two options:
  - A new table for the relationship
  - Add extra attribute(s) to “many” side

- New table
  - \textit{loan\_branch}(\textit{loan\_number}, \textit{branch\_name})

- Add an extra attribute to \textit{loan}
  - \textit{loan}=(\textit{loan\_number}, \textit{amount}, \textit{branch\_name})
Step 2.2: Many-to-Many Relation

- Two tables: attributes are primary keys of participating entities
  - borrower={customer_id, loan_number}
  - depositor={customer_id, account_number}
How about One-to-One relation?

- Merge two tables as one
Step 2: Relation

- Customer
  - customer_id
  - customer_name
  - customer_addr

- Borrow

- Loan
  - Loan_number
  - amount

- Branch
  - branch_name
  - branch_city
  - asset

- Account
  - account_num
  - balance

- Depositor

- Savings Account
  - interest_rate

- Checking Account
  - overdraft_amount
Step3: IS-A Relation

- IS-A: One class is a subclass of another class

- Two tables
  - savings_account=(account_number, interest_rate)
  - checking_account=(account_number, overdraft_amount)
Step 4: Identify primary key & foreign key

- **Entity**
  - Many-Many
  - Many-One
  - IS-A

- **customer**
  - `customer_id`
  - `customer_name`
  - `customer_addr`

- **borrow**
  - `customer_id`
  - `loan_number`

- **loan**
  - `loan_number`
  - `amount`
  - `branch_name`

- **depositor**
  - `customer_id`
  - `account_num`

- **account**
  - `account_num`
  - `account`
  - `branch_name`

- **branch**
  - `branch_name`
  - `branch_city`
  - `assets`

- **checking**
  - `account_num`
  - `interest_rate`

- **saving**
  - `account_num`
  - `overdraft_amount`
Revisit SQL

- When search multiple tables, we join they with the connection of their keys.

- Ex: What is the location of for account #1?
  - SELECT city from branch JOIN account ON branch.name = account.branch WHERE account.number = 1
Project

- Phase 1:
  - Write static html pages with functionalities and draw a diagram for the pages
  - Schedule demo and send the zip file to your TA
  - Deadline extend to Friday before your demo

- Example: Order pizza
Project

Diagram

- register.html
- login.html
- main.html
- Index.html

Register → Register

Order_success

login

submit
Primary key

- $\text{customer} = (\text{customer}\_id, \text{customer}\_name, \text{customer}\_addr)$
  - Two rows $t1, t2$ in the customer table
  - If $t1.\text{customer}\_id = t2.\text{customer}\_id$.
    - then $t1.\text{customer}\_name = t1.\text{customer}\_name$ AND $t1.\text{customer}\_addr = t2.\text{customer}\_addr$
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 $customer$