COMP163
Database Management Systems
September 9, 2008
Lecture 5 – Chapter 7
ER to Relational Schema Translation
NEXT UP

- skip ahead to Chapter 7:
  Translating ER Schemas to Relational Schemas
- then back to Chapter 6:
  The Relational Algebra: operations on relations
Specify all **foreign keys** for the following schema:

STUDENT(SID, Name, Major, BirthDate)

COURSE(CID, Name, Dept)

ENROLL(Student, Course, Semester, Grade)

BOOK_ADOPTION(Course, Semester, ISBN)

TEXTBOOK(ISBN, Title, Publisher, Author)
Reverse engineer this relational schema to find an equivalent ER schema.
EER Bank Schema
Step 1: Regular Entities

- Regular entity types become relations
  - include all simple attributes
  - include only components of compound attributes
  - keys become primary keys
  - if multiple keys (candidates) select a primary key

CUSTOMER(Ssn, Name, Addr, Phone)
Step 1: Regular Entities

BANK(\text{Code}, \text{Name}, \text{Addr})

ACCOUNT(\text{Acct\_no}, \text{Type}, \text{Balance})

LOAN(\text{Loan\_no}, \text{Type}, \text{Amount})
Step 2: Weak Entities

- Weak entity types become relations
  - include all simple attributes
  - include only components of compound attributes
  - create a primary key from partial key and key of owning entity type (through identifying relationship)
  - attributes acquired through identifying relationship become a foreign key*

* typically, deletions and insertions will be propagated through this foreign key
Step 2: Weak Entities

- Weak entity types become relations

```
BANK_BRANCH(Bank_code, Branch_No, Addr)
```

FK

```
BANK(Code, Name, Addr)
```
Step 3: Binary 1:1 Relationships

- **Approach 1: Foreign Key**
  - Chose one of the related entity types to hold the relationship (chose one with total participation, if possible)
  - add FK to other relation
  - move all relationship attributes to this relation
  - *this approach is preferable, except as noted below*

- **Approach 2: Merged Relation**
  - combine the relations for the related entities into a single relation
  - *use only when both participations are total*

- **Approach 3: Separate Relation**
  - same as binary M:N relationship (see step 5)
  - *not generally a good option*
Step 3: Binary 1:1 Relationships

- Approach 1: Foreign Key

```plaintext
EMPLOYEE(Ssn, Name, ...)

DEPARTMENT(Name, Number, Mgr, Mgr_start_date)
```
Step 3: Binary 1:1 Relationships

- Approach 2: Merged Relation

\[ AJB(x, y, p, q, r) \]

or

\[ AJB(x, y, p, q, r) \]
Step 4: Binary 1:N Relationships

- 1:N Relationships become foreign key at N side
  - any relationship attributes also go to N side

```
LOAN(Loan_no, Type, Amount, Bank, Branch)

BANK_BRANCH(Bank_code, Branch_No, Addr)
```
Step 4: Binary 1:N Relationships

- 1:N Relationships become foreign key at N side
  - any relationship attributes also go to N side

ACCOUNT\(\text{Acct\_no, Type, Balance, Bank, Branch}\) → BANK\_BRANCH\(\text{Bank\_code, Branch\_No, Addr}\)
Step 5: Binary M:N Relationships

- M:N Relationships must become a new relation
  - contains FKs to both related entities
  - combined FKs become PK for new relations
  - relationship attributes go in new relation

CUSTOMER(Ssn, Name, Addr, Phone)

A_C(Acct, Cust)

ACCOUNT(Acct_no, Type, Balance, Bank, Branch)
Step 6: Multivalued Attributes

- Multivalued attributes must become new relations
  - FK to associated entity type
  - PK is whole relation

DEPARTMENT(\text{Name, Number, Mgr, Mgr\_start\_date})

\text{DEPT\_LOCATIONS(DName, Dno, Location)}
Step 7: N-ary Relationships

- Non-Binary Relationships become new relations
  - FKS to all participating entity types
  - Combine FKS to make a PK (exclude entities with max participation of 1)
  - Include any relationship attributes

SUPPLIER(SName)

PROJECT(Proj_name)

PART(Part_no)

SUPPLY(SName, PName, Part, Quantity)
Completed Bank Schema

CUSTOMER(Ssn, Name, Addr, Phone)
BANK(Code, Name, Addr)
ACCOUNT(Account_no, Type, Balance, Bank, Branch)
LOAN(Loan_no, Type, Amount, Bank, Branch)
BANK_BRANCH(Bank_code, Branch_No, Addr)
A_C(Account, Customer)
L_C(Loan, Customer)

BANK_BRANCH(Bank_code) refers to BANK
LOAN(Bank,Branch) refers to BANK_BRANCH
ACCOUNT(Bank,Branch) refers to BANK_BRANCH
A_C(Account) refers to ACCOUNT
A_C(Customer) refers to CUSTOMER
L_C(Loan) refers to LOAN
L_C(Customer) refers to CUSTOMER
Bank Schema: MS Access
Step 8: Inheritance

- Option a: Each entity type becomes a relation
  - all have same PK (from superclass)
  - PKs in subclasses are FKs to superclass
  - *most general solution*

PERSON(\text{ID}, \text{Name})
STUDENT(\text{ID}, \text{Major}, \text{Class})
PROFESSOR(\text{ID}, \text{Dept}, \text{Office})

STUDENT(\text{ID}) refers to PERSON
PROFESSOR(\text{ID}) refers to PERSON
Step 8: Inheritance

- Option b: Each subclass becomes a relation
  - all have same PK (from superclass)
  - each relation gets all superclass attributes
  - restriction: only works for covering inheritance
  - problem: need to join tables to find all PERSONs

STUDENT (ID, Name, Major, Class)
PROFESSOR (ID, Name, Dept, Office)
Step 8: Inheritance

- Option c: Single relation with a type discriminator
  - PK from superclass
  - all attributes from all classes
  - restriction: only works for disjoint inheritance
  - problem: lots of NULL values

PERSON(ID, Classifier, Name, Major, Class, Dept, Office)

Classifier ∈ { ‘S’, ‘P’, ‘N’ }
Step 8: Inheritance

- Option d: Single relation with multiple discriminators
  - PK from superclass
  - all attributes from all classes
  - works for overlapping inheritance
  - problem: lots of NULL values

PERSON(ID, isStudent, isProfessor, Name, Major, Class, Dept, Office)

dom(isStudent) = Boolean
dom(isProfessor) = Boolean
Step 9: Unions

- Union types become a new relation of surrogate keys
  - surrogate keys are added to all defining classes
  - attributes of the union type go in the new relation

add surrogate key to OWNER
Step 9: Unions

PERSON(Driver_license_no, Ssn, Name, Address, Owner_id)
BANK(Bname, Baddress, Owner_id)
COMPANY(Cname, Caddress, Owner_id)
OWNER(ID)

PERSON(Owner_id) refers to OWNER
BANK(Owner_id) refers to OWNER
COMPANY(Owner_id) refers to OWNER
Step 9: Unions

add surrogate key to REGISTERED_VEHICLE
Step 9: Unions

CAR(Vehicle_id, Cstyle, Cmake, Cyear, Cmodel)
TRUCK(Vehicle_id, Tonnage, Tmake, Tyear, Tmodel)
REGISTERED_VEHICLE(Vehicle_id, License_plate_no)

CAR(Vehicle_id) refers to REGISTERED_VEHICLE
TRUCK(Vehicle_id) refers to REGISTERED_VEHICLE

in this case, we don’t need to invent a surrogate key, since the domains of
CAR keys and TRUCK keys are the same (and non-overlapping)
Step 9: Unions

OWNS relation uses the surrogate keys

OWNS(Owner_id, Vehicle_id,
    Purchase_date, Lien_or_regular)

OWNS(Owner_id) refers to OWNER
OWNS(Vehicle_id) refers to REGISTERED_VEHICLE
EXERCISES

- Create relational schema from the following:
Notes:
A LEG (segment) is a nonstop portion of a flight.
A LEG_INSTANCE is a particular occurrence of a LEG on a particular date.
### Employee Table

<table>
<thead>
<tr>
<th>Fname</th>
<th>Minit</th>
<th>Lname</th>
<th>Ssn</th>
<th>Bdate</th>
<th>Address</th>
<th>Sex</th>
<th>Salary</th>
<th>Super_ssn</th>
<th>Dno</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>B</td>
<td>Smith</td>
<td>123456789</td>
<td>1965-01-09</td>
<td>731 Fondren, Houston, TX</td>
<td>M</td>
<td>30000</td>
<td>333445555</td>
<td>5</td>
</tr>
<tr>
<td>Franklin</td>
<td>T</td>
<td>Wong</td>
<td>333445555</td>
<td>1955-12-08</td>
<td>638 Voss, Houston, TX</td>
<td>M</td>
<td>40000</td>
<td>888665555</td>
<td>5</td>
</tr>
<tr>
<td>Alicia</td>
<td>J</td>
<td>Zelaya</td>
<td>999887777</td>
<td>1968-01-19</td>
<td>3321 Castle, Spring, TX</td>
<td>F</td>
<td>25000</td>
<td>987654321</td>
<td>4</td>
</tr>
<tr>
<td>Jennifer</td>
<td>S</td>
<td>Wallace</td>
<td>987654321</td>
<td>1941-06-20</td>
<td>291 Berry, Bellaire, TX</td>
<td>F</td>
<td>43000</td>
<td>888665555</td>
<td>4</td>
</tr>
<tr>
<td>Ramesh</td>
<td>K</td>
<td>Narayan</td>
<td>6668844444</td>
<td>1962-09-15</td>
<td>975 Fire Oak, Humble, TX</td>
<td>M</td>
<td>38000</td>
<td>333445555</td>
<td>5</td>
</tr>
<tr>
<td>Joyce</td>
<td>A</td>
<td>English</td>
<td>453453453</td>
<td>1972-07-31</td>
<td>5631 Rice, Houston, TX</td>
<td>F</td>
<td>25000</td>
<td>333445555</td>
<td>5</td>
</tr>
<tr>
<td>Ahmadd</td>
<td>V</td>
<td>Jabbar</td>
<td>987987987</td>
<td>1969-03-29</td>
<td>980 Dallas, Houston, TX</td>
<td>M</td>
<td>25000</td>
<td>987654321</td>
<td>4</td>
</tr>
<tr>
<td>James</td>
<td>E</td>
<td>Borg</td>
<td>888665555</td>
<td>1937-11-10</td>
<td>450 Stone, Houston, TX</td>
<td>M</td>
<td>55000</td>
<td>NULL</td>
<td>1</td>
</tr>
</tbody>
</table>

### Department Table

<table>
<thead>
<tr>
<th>Dname</th>
<th>Dnumber</th>
<th>Mgr_ssn</th>
<th>Mgr_start_date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Research</td>
<td>5</td>
<td>333445555</td>
<td>1988-05-22</td>
</tr>
<tr>
<td>Administration</td>
<td>4</td>
<td>987654321</td>
<td>1995-01-01</td>
</tr>
<tr>
<td>Headquarters</td>
<td>1</td>
<td>888665555</td>
<td>1981-06-19</td>
</tr>
</tbody>
</table>

### DEPT_LOCATIONS

<table>
<thead>
<tr>
<th>Dnumber</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Houston</td>
</tr>
<tr>
<td>4</td>
<td>Stafford</td>
</tr>
<tr>
<td>5</td>
<td>Bellaire</td>
</tr>
<tr>
<td>5</td>
<td>Sugarland</td>
</tr>
<tr>
<td>5</td>
<td>Houston</td>
</tr>
</tbody>
</table>

### Works_ON

<table>
<thead>
<tr>
<th>Essn</th>
<th>Pno</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>123456789</td>
<td>1</td>
<td>32.5</td>
</tr>
<tr>
<td>123456789</td>
<td>2</td>
<td>7.5</td>
</tr>
<tr>
<td>6668844444</td>
<td>3</td>
<td>40.0</td>
</tr>
<tr>
<td>453453453</td>
<td>1</td>
<td>20.0</td>
</tr>
<tr>
<td>453453453</td>
<td>2</td>
<td>20.0</td>
</tr>
<tr>
<td>333445555</td>
<td>2</td>
<td>10.0</td>
</tr>
<tr>
<td>333445555</td>
<td>3</td>
<td>10.0</td>
</tr>
<tr>
<td>333445555</td>
<td>10</td>
<td>10.0</td>
</tr>
<tr>
<td>333445555</td>
<td>20</td>
<td>10.0</td>
</tr>
<tr>
<td>999887777</td>
<td>30</td>
<td>30.0</td>
</tr>
<tr>
<td>999887777</td>
<td>10</td>
<td>10.0</td>
</tr>
<tr>
<td>987987987</td>
<td>10</td>
<td>35.0</td>
</tr>
<tr>
<td>987987987</td>
<td>30</td>
<td>5.0</td>
</tr>
<tr>
<td>987654321</td>
<td>30</td>
<td>20.0</td>
</tr>
<tr>
<td>987654321</td>
<td>20</td>
<td>15.0</td>
</tr>
<tr>
<td>888665555</td>
<td>20</td>
<td>NULL</td>
</tr>
</tbody>
</table>

### Project Table

<table>
<thead>
<tr>
<th>Pname</th>
<th>Pnumber</th>
<th>Location</th>
<th>Dnum</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProductX</td>
<td>1</td>
<td>Bellaire</td>
<td>5</td>
</tr>
<tr>
<td>ProductY</td>
<td>2</td>
<td>Sugarland</td>
<td>5</td>
</tr>
<tr>
<td>ProductZ</td>
<td>3</td>
<td>Houston</td>
<td>5</td>
</tr>
<tr>
<td>Computerization</td>
<td>10</td>
<td>Stafford</td>
<td>4</td>
</tr>
<tr>
<td>Reorganization</td>
<td>20</td>
<td>Houston</td>
<td>1</td>
</tr>
<tr>
<td>Newbenefits</td>
<td>30</td>
<td>Stafford</td>
<td>4</td>
</tr>
</tbody>
</table>

### DEPENDENT

<table>
<thead>
<tr>
<th>Essn</th>
<th>Dependent_name</th>
<th>Sex</th>
<th>Bdate</th>
<th>Relationship</th>
</tr>
</thead>
<tbody>
<tr>
<td>333445555</td>
<td>Alice</td>
<td>F</td>
<td>1986-04-05</td>
<td>Daughter</td>
</tr>
<tr>
<td>333445555</td>
<td>Theodore</td>
<td>M</td>
<td>1983-10-25</td>
<td>Son</td>
</tr>
<tr>
<td>333445555</td>
<td>Joy</td>
<td>F</td>
<td>1958-05-03</td>
<td>Spouse</td>
</tr>
<tr>
<td>987654321</td>
<td>Abner</td>
<td>M</td>
<td>1942-02-28</td>
<td>Spouse</td>
</tr>
<tr>
<td>123456789</td>
<td>Michael</td>
<td>M</td>
<td>1988-01-04</td>
<td>Son</td>
</tr>
<tr>
<td>123456789</td>
<td>Alice</td>
<td>F</td>
<td>1988-12-30</td>
<td>Daughter</td>
</tr>
<tr>
<td>123456789</td>
<td>Elizabeth</td>
<td>F</td>
<td>1967-05-05</td>
<td>Spouse</td>
</tr>
</tbody>
</table>
Preview: Queries

- Which employee has no supervisor?
- Which employees are supervised by Franklin Wong?
- Which employees have dependents?
- Which employees have daughters?
- Which employees in department 5 work more than 10 hours/week on ProductX?
- Which department has the highest paid manager?
- What is the average salary of all department managers?
- Which employees don’t have daughters?