ISYS2120
Feedback on Asst1

Alan Fekete

Sem2, 2022

Marks from Asst1

- All submissions should be marked in Canvas
  - Email alan.fekete@Sydney.edu.au if you can’t see your mark – name the group involved, and include ISYS2120 in email subject line
  - Also email if you don’t understand reasons you lost marks
- Recall, submission is marked out of 4; all members get the same
  - Mark is based on the criteria, not on counting correct aspects etc – eg even a single mistake that means your design can’t always capture most of information from an arbitrary state of the domain, results in score of 1 out of 4
- Also, you have up to 0.5 point for individual progress in each of week 3 and 4

Overall structure – Entity Sets

- Usually, CREATE one relational table for each Entity Set
  - Some choice of what type for each column (many will be VARCHAR(n) for some suitable max length n, but maybe use INTEGER for IDs, DATE for dates, FLOAT for measurement such as weight, FLOAT(2) for money etc)
  - Some choice about whether a column will be declared NOT NULL
- This was mostly done well
- Some tricky cases though, where some groups had a design which would not be able to capture an arbitrary state of the domain that would fit the diagram

Multivalued attributes

- Eg Department has attribute OfficeLocations
  - The recommended approach is to declare a separate table for this

```
CREATE TABLE OfficeLocations (
    Name VARCHAR(15) FOREIGN KEY REFERENCES Department(Name),
    Office VARCHAR(20)
    PRIMARY KEY (Name, Office)
)
```

  - Not correct: if you make Name the primary key [because then you can’t capture information about a department with more than one officelocation]
  - Not correct: if you put OfficeLocations as a column in the Office table [either you have only one location there, or the column contains some string which internally lists all the locations, but in the latter design, it is hard to extract the individual locations for joins etc]
- Similar for Employee with multivalued PhoneNumbers

Composite Primary Key

- Where the diagram shows a composite primary key, the table must also have this
  - Eg Model has primary key (ModelNumber, Manufacturer)
- This combination can be used as foreign key to capture relationship sets
- Eg For InstanceOf

```
CREATE TABLE Device (
   DeviceID INTEGER PRIMARY KEY,
….
   InstanceOfModelNumberVARCHAR(10),
   InstanceOfManufacturer VARCHAR(15),
   FOREIGN KEY (InstanceOFModelNumber, InstanceOfManufacturer) REFERENCES
Model (ModelNumber, Manufacturer)
)
```

Inheritance

- Information about each Phone
  - Two possible design styles: as separate Phone table, or putting information in Device table (with NULL entries for those that are not phones)
  - If there is a separate Phone table, it must have a column for the DeviceID, and this would be the PRIMARY KEY in Phone, and also a FOREIGN KEY to the Device table
  - If the information about a Phone (Plan, Carrier, Number) is in Device table, these columns must be allowed to be NULL

Overall structure – Relationship Sets

- For many-many relationships, one must CREATE an extra table for the relationship
  - And ensure the PRIMARY KEY allows for the many cardinality cases
- Eg Device UsedBy Employee

```
CREATE TABLE UsedBy (
    DeviceID INTEGER FOREIGN KEY REFERENCES Device(DeviceID),
    EmpID INTEGER FOREIGN KEY REFERENCES Employee(EmpID),
    PRIMARY KEY (DeviceID, EmpID)
)
```

Many-One relationship Sets

- For many-one relationship types, there is a choice between CREATE an extra table, or put a foreign key column in the many side
  - Do not make the foreign key column NOT NULL, except if you know there is total participation on the many side
- Eg Device IssuedTo Employee

```
CREATE TABLE Device (
   DeviceID INTEGER PRIMARY KEY,
  ….
   IssuedTo INTEGER FOREIGN KEY REFERENCES Employee(EmpID)
)
Or
CREATE TABLE IssuedTo (
    DeviceID INTEGER PRIMARY KEY FOREIGN KEY REFERENCES Device(DeviceID),
    EmpID INTEGER FOREIGN KEY REFERENCES Employee(EmpID),
)
```

Constraints

- Cardinality Constraints
  - Many-one can be captured in the overall structure if you do it as a column in the table of the many side entity set
  - if instead you have a separate table for the relationship, the constraint is captured by the primary key being just the ID from the many side
- Participation constraints
  - Total participation by the many side in many-one, can be captured if you use a column in the table of the many side: make the foreign key column NOT NULL
  - Other cases are rather hard to capture in the schema
    - Could use multi-table assertion [but this is not supported in PostgreSQL]