

CSE 444 Midterm Exam

October 29, 2007

Name Sample Solution

Question 1	/ 40
Question 2	/ 30
Question 3	/ 30
Total	/ 100

Question 1. SQL (40 points) Consider the following schema for a database that contains information about books, authors, and publishers.

BOOK (isbn, title, publisher, year)

AUTHOR (ssn, name, city)

PUBLISHER (name, city)

WROTE (isbn, ssn)

Every book has a unique isbn (standardized book number) and every author has a unique ssn. To keep the problem simpler, we will assume that authors have single names (like “Cher”, “Bono”, or “Knuth”), and we will also assume that each publisher has a unique name and is located in a single city. The WROTE relationship connects author ssns with the isbns of books that they wrote, either as a single author or as one of several co-authors.

(a) Write a SQL query that gives the titles and years of all books written by the author named Rowling.

```
SELECT b.title, b.year
FROM Book b, Author a, Wrote w
WHERE b.isbn = w.isbn
      AND w.ssn = a.ssn
      AND a.name = 'Rowling';
```

(continued next page)

(b) Write a SQL query that gives the unique names of all authors who have written a book where one of the book's authors has the name Ullman (i.e., Ullman and all of his co-authors on all of his books).

```
SELECT DISTINCT coauthors.name
FROM Author a, Wrote w, Wrote coauthorsWrote, Author coauthors
WHERE a.name = 'Ullman'
      AND a.ssn = w.ssn
      AND w.isbn = coauthorsWrote.isbn
      AND coauthorsWrote.ssn = coauthors.ssn;
```

(c) Write a SQL query that lists the names of all publishers in the city of Boston and the number of books that they published in the decade beginning in 1990 and ending in 1999.

```
SELECT p.name, COUNT(*)
FROM Publisher p LEFT OUTER JOIN Book b ON p.name = b.publisher
WHERE b.year BETWEEN 1990 AND 1999
      AND p.city = 'Boston'
GROUP BY p.name;
```

(continued next page)

(d) Write one or more SQL statements to add the following book to this database:

Algorithms for Smarties by Knuth, Addison-Wesley, 1968, ISBN = 0-321-1742-X

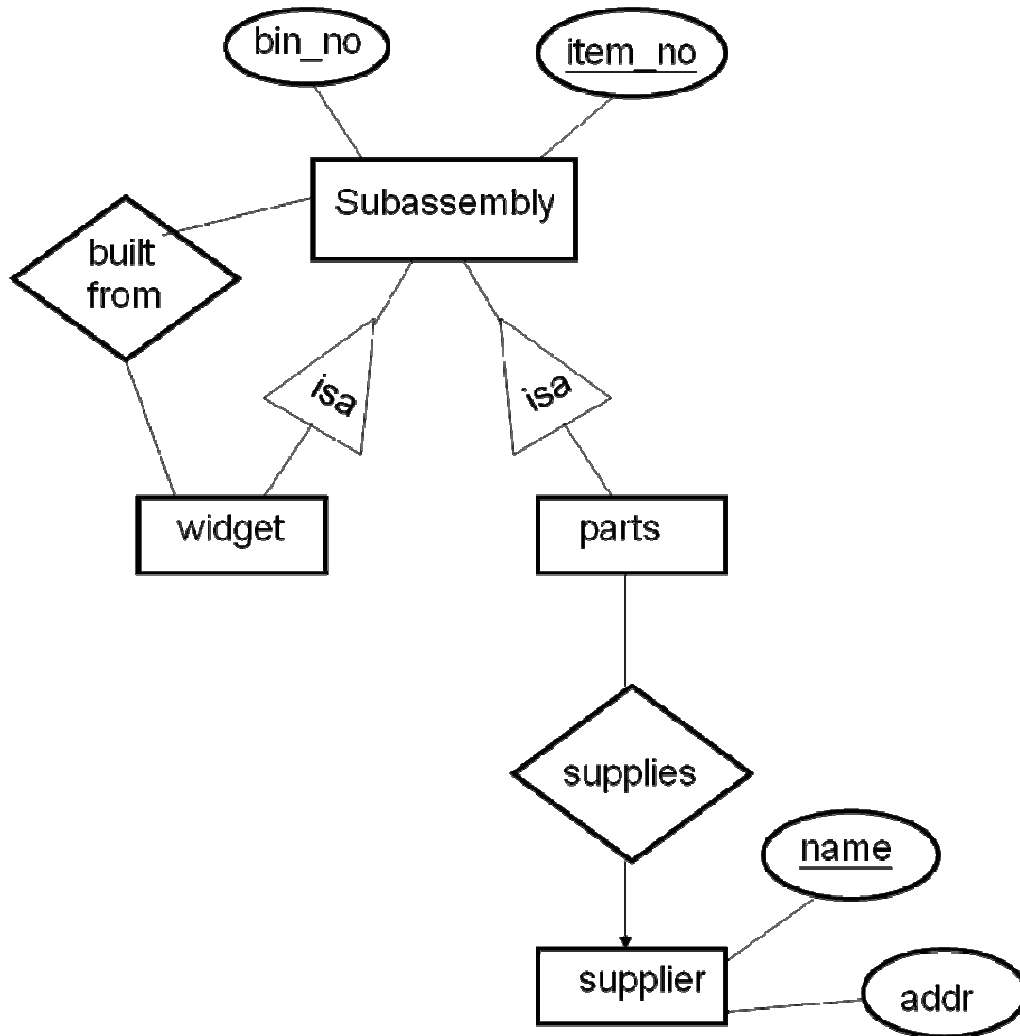
You may assume that there already are entries for Knuth in the AUTHOR table and for Addison-Wesley in the PUBLISHER table, and you may assume that there is only one author named Knuth in the AUTHOR table.

```
INSERT INTO Book(isbn, title, publisher, year)
VALUES('0-321-1742-X', 'Algorithms for Smarties', 'Addison-Wesley', 1968);
```

```
INSERT INTO Wrote(isbn, ssn)
  SELECT '0-321-1742-X', ssn
  FROM Author
  WHERE name = 'Knuth';
```

Question 2. E/R Diagrams (30 points) Wanda’s Widget Works (aka WWW, Inc.) needs your help designing a database to keep track of the flow of parts in the factory. Every part has an item number, a supplier name, and a bin number where it is kept in the factory. Widgets are built from parts and also from other widgets, which are used as sub-assemblies. Each widget has an item number and a bin number, just like a part. Part suppliers have names and addresses and, to keep the problem simple, we will assume that supplier names are unique and addresses are a single string.

(a) Give an E/R diagram for this application. State any assumptions you need to make.



(continued next page)

(b) Write appropriate CREATE TABLE statements for SQL tables to store the information in your E/R diagram from part (a). You should choose very simple atomic datatypes for the attributes. Indicate all keys, and include any foreign key constraints needed to preserve the integrity of the data.

```
CREATE TABLE SubAssembly
(
  item_no INT PRIMARY KEY
  bin_no INT NOT NULL
);
```

```
CREATE TABLE Supplier
(
  name varchar(125) PRIMARY KEY,
  addr varchar(max) NOT NULL
);
```

```
CREATE TABLE Widget
(
  item_no INT PRIMARY KEY REFERENCES SubAssembly(item_no)
);
```

```
CREATE TABLE Part
(
  item_no INT PRIMARY KEY REFERENCES SubAssembly(item_no),
  supplier_id INT REFERENCES Supplier(name)
);
```

```
CREATE TABLE Built_From
(
  widget_no INT REFERENCES Widget(item_no),
  component INT REFERENCES SubAssembly(item_no)
);
```

Question 3. (30 points) Some of your old high-school friends have opened a new restaurant, and it has become so wildly successful that they need a computer to keep track of dinner reservations. Not knowing all that much about databases, they have created a single table to hold reservation information:

RESERVATION (Date, Time, Name, Phone, VIP)

Some customers have only made a single reservation, but many of them have multiple reservations in the table. No two customers have the same name and phone number, but some different customers have either the same name or same phone number (but not both). VIP is a boolean that is true to indicate the very best customers, who receive extra special service.

(a) What are the possible key(s) and superkeys(s) for this relation? Which of these possibilities form a minimal key for this relation? Justify your answer in terms of functional dependencies and closures.

The only non-trivial functional dependency in the table is Name Phone \rightarrow VIP. Both of the sets {Date,Time,Name,Phone,VIP} and {Date,Time,Name,Phone} are superkeys, because their closures include all of the attributes, but only {Date,Time,Name,Phone} is a minimal key.

(continued next page)

(b) Identify any “bad” functional dependencies in the RESERVATION table and use them to decompose it into relations that are in Boyce-Codd Normal Form (BCNF). The resulting relations should retain all of the information in the original table (i.e., it should be a “lossless” decomposition).

The non-trivial F.D. Name Phone \rightarrow VIP violates BCNF because {Name,Phone} is not a superkey for the RESERVATION relation. So we decompose the original RESERVATION table into the following two tables:

RESERVATION (Date, Time, Name, Phone)

CUSTOMER (Name, Phone, VIP)

Both of these relations are in BCNF and they contain all of the information in the original table.