

Database Design, CSCI 340, Spring 2015
Final, May 12

Multiple-Choice

1. Which of the following is not part of the vocabulary of database keys? (3 pts.)
- a. Referential key
 - b. Composite key
 - c. Primary key
 - d. Candidate key
 - e. Foreign key

2. Indexes created on a relation _____ writes and _____ reads. (3 pts.)
- a. speed up, speed up
 - b. slow down, slow down
 - c. speed up, slow down
 - d. slow down, speed up
 - e. have no effect

3. When you want to see all of the awards, even those not yet granted to a student, replace JOIN in the following

Student JOIN Award ON Student.studentID=Award.studentID

with: (3 pts.)

- a. LEFT OUTER JOIN
 - b. RIGHT OUTER JOIN
 - c. THETA JOIN
 - d. EQUI JOIN
 - e. JOIN (i.e. no change)
4. Relational DBMS are designed for _____ scaling, while NoSQL products are designed for _____ scaling. (3 pts.)
- a. up, out
 - b. out, in
 - c. wide, deep
 - d. horizontal, vertical
 - e. vertical, horizontal
5. Which of the following is NOT a characteristic of big data? (3 pts.)
- a. variety
 - b. simplicity
 - c. velocity
 - d. variability
 - e. volume

6. Which of the following is NOT an advantage of XML? (3 pts.)
- extensibility
 - separation of content and presentation
 - flexibility
 - integrity
 - simplicity
7. Normalization is (3 pts.)
- a process of transforming a relation into a set of relations with desirable properties
 - a process in which data attributes within a data model are organized to increase the cohesion of entity types
 - a process to reduce and even eliminate data redundancy
 - none of the above
 - all of the above
8. In relational algebra the function of the select symbol (σ) most closely corresponds to which SQL keyword? (3 pts.)
- SELECT
 - FROM
 - WHERE
 - ORDER BY
 - GROUP BY
9. Sometimes a relation is denormalized because denormalization will: (3 pts.)
- speed up database reads
 - speed up database writes
 - increase the integrity of the data
 - none of the above
 - all of the above
10. Say that a query performs the relational algebra operations of 'projection', 'selection' and 'join'. When possible, the query optimizer would use the order: (3 pts.)
- First join, then selection, finally projection
 - First join, then projection, finally selection
 - First selection, then join, finally projection
 - First projection, then join, finally selection
 - First selection, then projection, finally join

Short Answer

11. Consider the following relation.

Employee	
PK	<u>employeeNo</u>
	fName lName street city state zip

a. Is the relation in BCNF? Why or why not?

(2 pts.)

No because it contains the hidden FDs
 $zip \rightarrow city, state$

b. Describe all advantages and all disadvantages of the design of the Employee relation.

(3 pts.)

The above denormalized design will speed up queries for the address of an employee because it won't be necessary to join two tables together.

Having a separate zip table would simplify the process of entering records into the address table. The city and state would not need to be included, knowing the zip code would be enough. Also, if the zip codes change, it wouldn't require making city and state changes in several tables. Also, if no employee lives at a certain zip code, I'd still know the city and state associated with that zip code.

12. Discuss the consequences of data retention policies on privacy. (5 pts.)

The longer data is retained, the more likely that it is:

- no longer well maintained
- out dated
- can be compromised

This risk of privacy violation is increasing as data mining techniques become more sophisticated.

Data retention policies define parameters of what data is kept, how it is kept, where it is kept, and how long it is kept. These policies also define why the data is kept.

Some policies are in place to keep data such as medical records and academic records from being compromised. The EU Directive contains a comprehensive approach to data retention which protects Europeans. The U.S. does not have a comprehensive policy. Without a policy directing government agencies and private companies on the how long they are allowed to keep data, private citizens are at a great risk of having their privacy violated.

13.

- a. Describe the purpose of transactions. (5 pts.)

Some database updates require multiple changes. Transactions support this. Transactions are all or nothing. Either the whole transaction occurs, or none of it occurs. For instance, when transferring money from one account to another, it is important that the first balance is decremented and the second is incremented. Don't want just half of the transaction to be completed.

- b. List and describe the ACID properties (5 pts.)

A

Atomicity – Either the whole transaction is successful or none is.

C

Consistency – Takes the database from one consistent state to another.

I

Isolation – When one or more transactions run concurrently, their effects must be isolated from each other.

D

Durability- once a transaction is committed the results are permanent, that is, even if the database crashes, the recovery system will redo the transaction

Problem Solving

14. The relation:

sId	abbrev	pID	fName	lName	title	composer	years	dateTime	song
1	Leben	1	Billie	Meyer	Austria My Home	Schubert	1797-1828	2013-12-22 19:00:00	Lebenslust
3	Kyrie	1	Rich	Myer	Austria My Home	Beethoven	1770-1827	2013-12-22 19:00:00	Kyrie
3	Maria	1	Rich	Myer	Austria My Home	Schubert	1797-1828	2013-12-22 19:00:00	Ave Maria
3	Kyrie	2	Rich	Myer	Spring in the Air	Beethoven	1770-1827	2013-05-12 19:00:00	Kyrie
4	Leben	2	Jenn	Swartz	Spring in the Air	Schubert	1797-1828	2013-05-12 19:00:00	Lebenslust
5	Maria	2	Rich	Vandick	Spring in the Air	Schubert	1797-1828	2013-05-12 19:00:00	Ave Maria
2	Maria	3	Butch	Lee	Snowy Christmas	Schubert	1797-1828	2014-12-24 19:00:00	Ave Maria
4	Leben	3	Jenn	Swartz	Snowy Christmas	Schubert	1797-1828	2014-12-24 19:00:00	Lebenslust

8 rows in set (0.00 sec)

contains the FDs:

sId → fName, lName

pID → title, dateTime

abbrev → composer, years, song

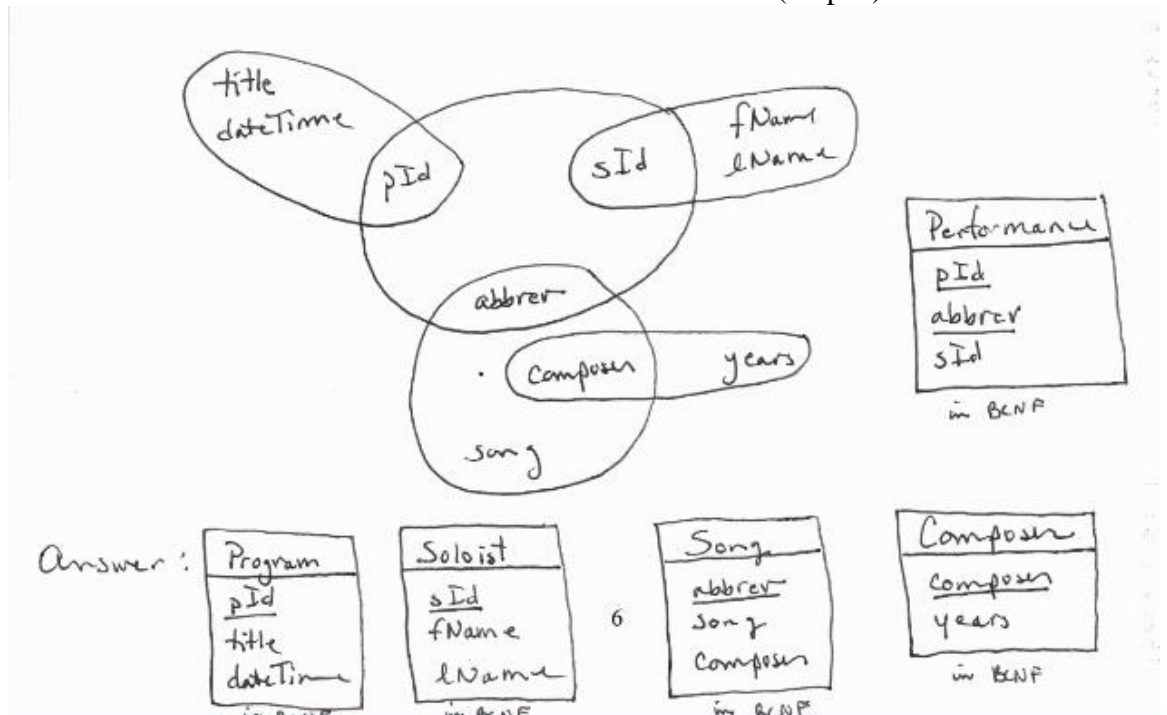
song → abbrev, composer, years

composer → years

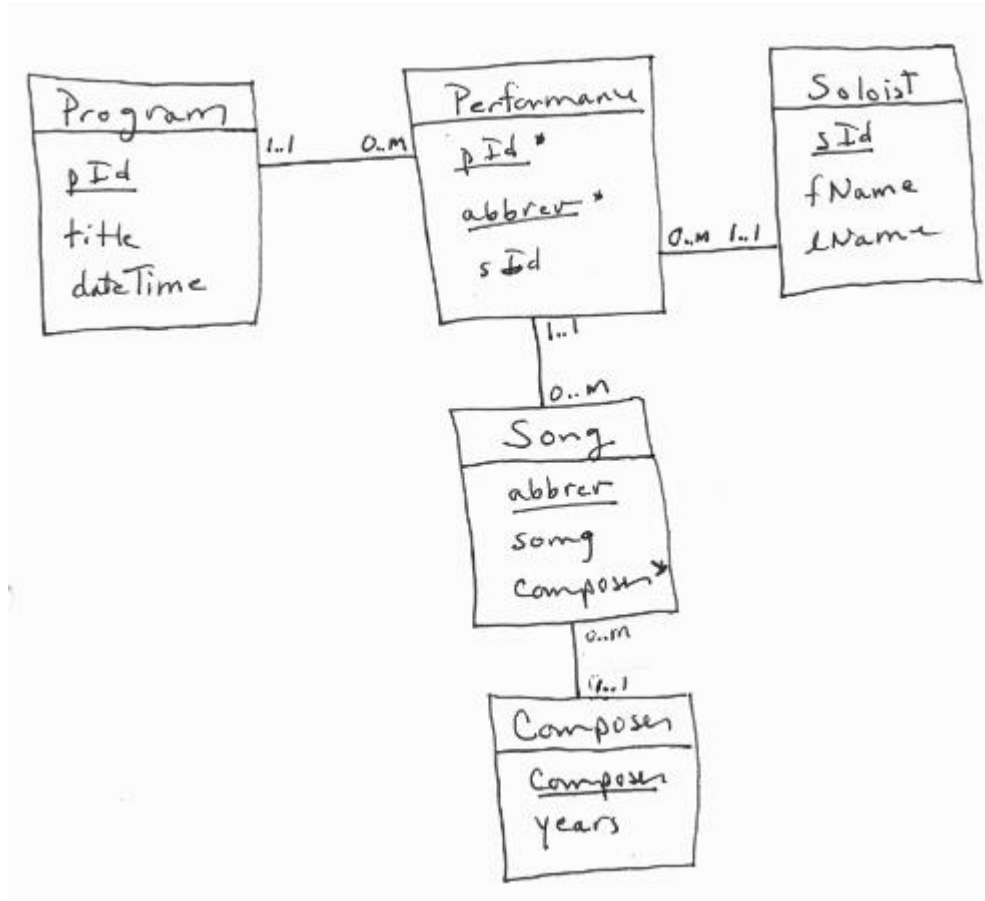
pId, abbrev → sId, fName, lName, title, composer, years, dateTime, song

Normalize the relation to a set of BCNF relations.

(10 pts.)



Another way to see the above answer:



15. The Friendly School Recruiting Office wants to record information about interactions that occur between someone interested in attending the school and recruiting office personnel. The recruiting office does not want to ask potential students lots of questions when they first inquire about the school, so they simply get the first and last name of the inquirer and assign that person a unique identification number. (The potential student never needs to be aware that the school has assigned them an identification number.) As more information about the potential student becomes available, such as the address (street, city, state, country, zip/country code), one or more phone numbers, and one or more email addresses, this information is recorded into the database. (Note that a recruit will only have one address.)

The system should be able to associate any number of phone numbers with the recruit and any number of email addresses. Identifiers such as “cell”, “home” or “preferred” should be able to be associated with each phone number. It should be possible to indicate which of the email addresses is preferred. The Friendly School Recruiting Office assumes, perhaps naively, that no two recruits will have the same phone number or email address.

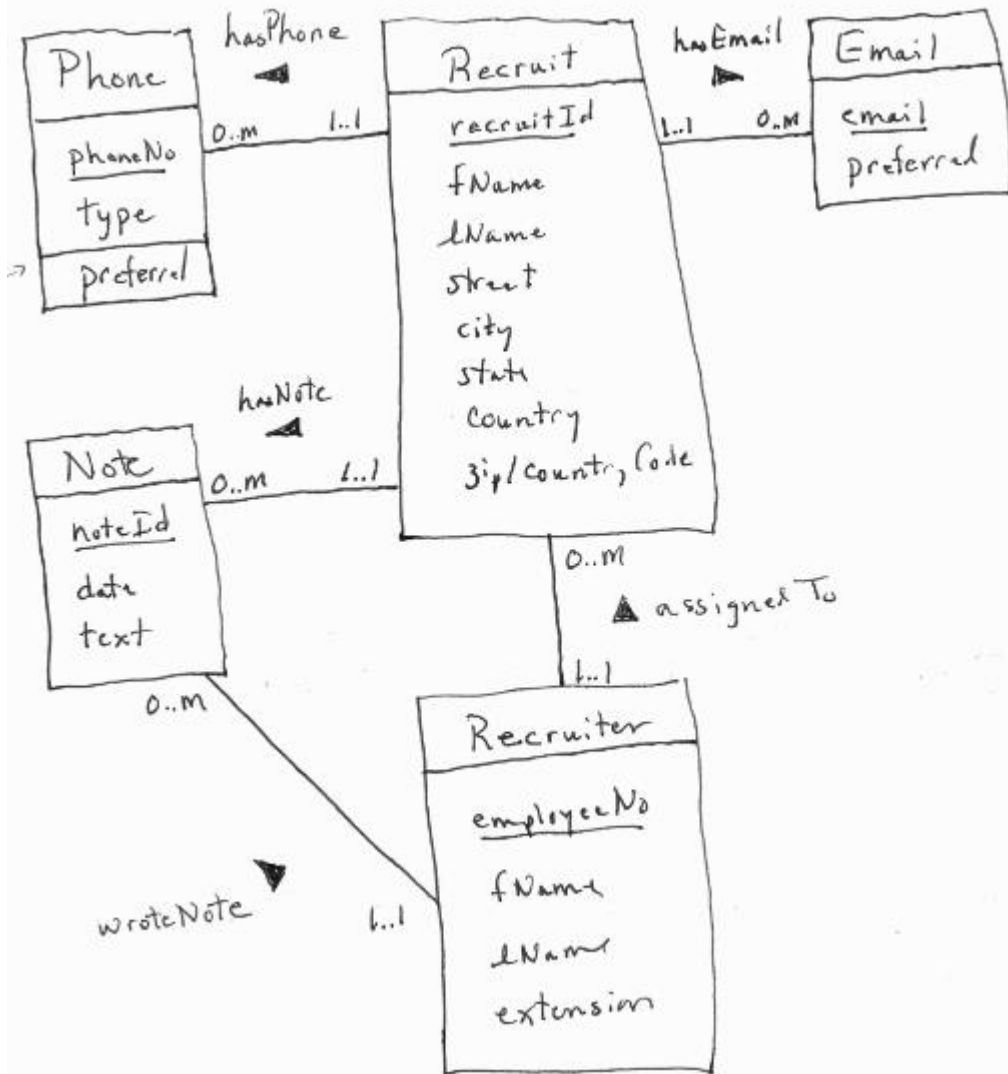
A recruiter is assigned to every potential student. This recruiter has responsibility for doing follow up work, contacting the potential student, answering their questions, and keeping the student aware of what needs to be done to apply to the school. In some cases the recruiter assigned to a potential student changes. The database only keeps track of the current recruiter.

An employee identification number, first and last name, and the telephone extension are kept for each recruiter. A recruiter may not have any students assigned to them, but typically he/she is assigned to several students.

Recruiting personnel often talk with many potential students so they want the system to allow them to record notes about a potential student. Every note should be dated and should indicate what recruiter made the note.

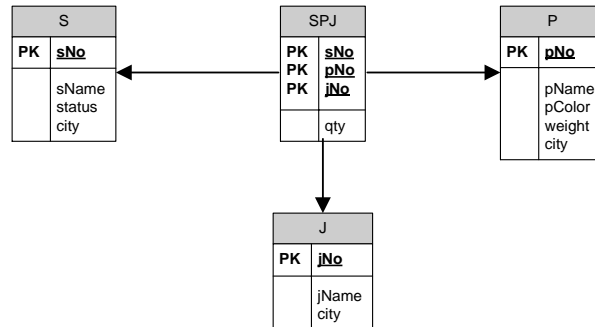
Imagine that you are responsible for developing this database application. Begin by developing a conceptual model for it. This model should:

- underline primary keys, (for partial primary keys use {PPK},
- include relationships with minimum and maximum cardinalities
- name the relationships indicating how to interpret the name via a triangular arrows ► and ◄.
- do not include multi-valued attributes in the design (10 pts)



Queries (written without a computer)

Suppliers-Parts-Project
Database



16.

- a. Write a relational algebra query which displays the name and status of all suppliers which supply parts for a project in Paris. (3 pts.)

$$\pi_{sName, status} \left(\left(S \bowtie SPJ \right) \bowtie \left(\sigma_{city='Paris'} J \right) \right)$$

or

$$\pi_{sName, status} \left(\sigma_{J.city='Paris'} \left(\left(S \bowtie SPJ \right) \bowtie J \right) \right)$$

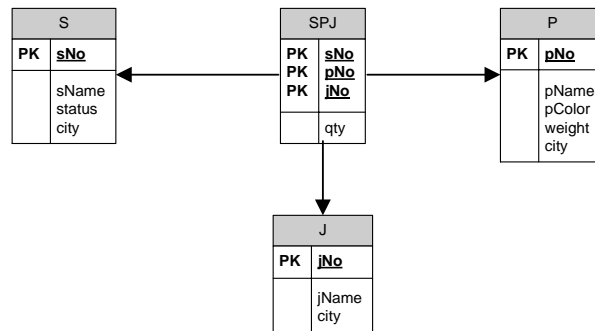
- b. Write this query in SQL.

(2 pts.)

```

SELECT sName, status
FROM S JOIN SPJ ON S.sNo=SPJ.sNo
      JOIN J ON SPJ.jNo=J.jNo
WHERE J.city='Paris';
    
```

Suppliers-Parts-Project
Database



17.

- a. Write a relational algebra query which lists every supplier name in the S table (whether or not the supplier appears in the SPJ table), followed by the total number of any parts which are supplied to any project. (3 pts.)

$$\pi_{sName, \sum(qty)} (S \times_{SPJ})$$

- b. Write this query in SQL. (2 pts.)

```
SELECT sName, IFNULL(SUM(qty),0)
FROM S LEFT JOIN SPJ ON S.sNo=SPJ.sNo
GROUP BY sName;
```

18. Say that the SPJ table holds the following.

sNo	pNo	jNo	qty
S001	P01	J001	200
S001	P01	J004	700
S002	P03	J001	400
S002	P03	J002	200
S002	P03	J003	200
S002	P03	J004	500
S002	P03	J005	600
S002	P03	J006	400
S002	P03	J007	800
S003	P03	J001	200
S003	P05	J002	100
S004	P06	J007	300
S005	P01	J004	100
S005	P01	J005	300
S005	P02	J002	200
S005	P02	J004	100
S005	P03	J004	200
S005	P04	J004	400
S005	P05	J004	500
S005	P05	J005	500
S005	P05	J007	100
S005	P06	J002	200

Note that supplier 'S002' supplies part 'P03' to projects 'J006' (quantity 400) and 'J007' (quantity 800). Project 'J006' needs more of part 'P03' so it has been decided that 200 parts which were going to be delivered to project 'J007' will now be delivered to project 'J006'. Write a transaction to accomplish this. (5 pts.)

```
BEGIN WORK;
```

```
UPDATE SPJ
```

```
SET qty=qty+200
```

```
WHERE sNo='S002' AND pNo='P03' AND jNo='J006';
```

```
UPDATE SPJ
```

```
SET qty=qty-200
```

```
WHERE sNo='S002' AND pNo='P03' AND jNo='J007';
```

```
COMMIT;
```

**Database Design, CSCI 340, Spring 2015
Final, Computer Portion, May 12**

Please turn in the first portion of the exam before beginning this portion. To complete this portion of the exam you may use your notes, any previous assignments, the text and/or the Internet. You may not communicate with anyone other than myself during this exam.

Login to your katie account and then to your mysql class account:

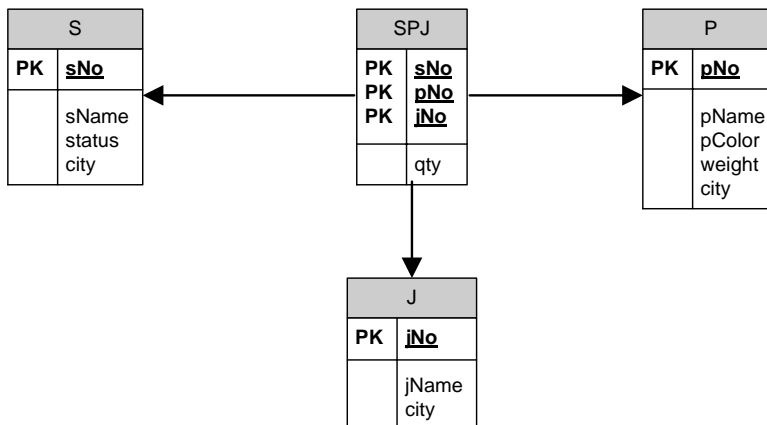
- mysql -u userName -p
- Enter password: password

Switch into the SuppliersPartsProjets database via:

- USE yourDBName;

Write queries for the following. Remember that your query must work for any extension of the database, not just the current extension.

Suppliers-Parts-Project
Database



19. Write an SQL query which displays the name and status of all suppliers which supply parts for a project in Paris. (5 pts.)

```
SELECT sName, status
FROM S JOIN SPJ ON S.sNo=SPJ.sNo
      JOIN J ON SPJ.jNo=J.jNo
WHERE J.city='Paris';
```

20. Write an SQL query which lists every supplier name in the S table (whether or not the supplier appears in the SPJ table), followed by the number of any parts which are supplied to any project. (5 pts.)

```
SELECT sName, IFNULL(SUM(qty),0)
FROM S LEFT JOIN SPJ ON S.sNo=SPJ.sNo
GROUP BY sName;
```

21. Say that the SPJ table holds the following.

sNo	pNo	jNo	qty
S001	P01	J001	200
S001	P01	J004	700
S002	P03	J001	400
S002	P03	J002	200
S002	P03	J003	200
S002	P03	J004	500
S002	P03	J005	600
S002	P03	J006	400
S002	P03	J007	800
S003	P03	J001	200
S003	P05	J002	100
S004	P06	J007	300
S005	P01	J004	100
S005	P01	J005	300
S005	P02	J002	200
S005	P02	J004	100
S005	P03	J004	200
S005	P04	J004	400
S005	P05	J004	500
S005	P05	J005	500
S005	P05	J007	100
S005	P06	J002	200

Note that supplier 'S002' supplies part 'P03' to projects 'J006' (quantity 400) and 'J007' (quantity 800). Project 'J006' needs more of part 'P03' so it has been decided that 200 parts which were going to be delivered to project 'J007' will now be delivered to project 'J006'. Write a transaction to accomplish this. (5 pts.)

```
BEGIN WORK;
```

```
UPDATE SPJ
```

```
SET qty=qty+200
```

```
WHERE sNo='S002' AND pNo='P03' AND jNo='J006';
```

```
UPDATE SPJ
```

```
SET qty=qty-200
```

```
WHERE sNo='S002' AND pNo='P03' AND jNo='J007';
```

```
COMMIT;
```