1.264 Final Exam
Fall, 2003

Exam guidelines:

1. Three hours are allowed to complete the exam.
2. If there are questions or clarifications, ask the proctor.
3. Write your answers in the space provided on the examination. Use the back of a page or insert extra pages if you need the space.
4. Open notes.
5. No books allowed.
6. No laptop computers allowed.
7. No cellphones or messaging devices are allowed. Please turn off any that you have brought. If you do not, you will be asked to leave and not allowed to complete the exam.
8. Calculators are not necessary and are not allowed. You will have to do some calculations in problem 5 (sorry—some of you might like a calculator for it) but you can do them very approximately.
9. 100 points total (20 points for each of 5 questions).
10. Put your name and email address on the exam booklet.
11. Short answer questions: Your answers are limited to a maximum of 2-4 sentences or phrases. Demonstrate that you understand the principles and key points. You will receive full credit for an answer if you make the principal observation(s) that the question is asking for. Details are not necessary.

1. **Software process. (20 points)**

After several attempts to resolve recording industry issues with MIT students sharing music files, MIT initiated its LAMP music project. It named Jane the project manager; she had several failed projects in her past, so she brought in a software process consultant, Erica, to help.

Erica met with Jane several times and recommended a skunkworks team. "Software people are creative and need a lot of freedom. Put them off-campus, and give them lots of autonomy. They'll work day and night and will complete the project on time." Jane was uncomfortable with an off-campus office but she took Erica's advice. She put her best developer, Ann, in charge.

"Ann, we need to build LAMP as fast as possible before anyone goes to jail. There has been a student group that has drafted what they'd like the system to do. I've looked at this —it's sort of a requirements document—and it looks good. We need to get LAMP out in 3 months. Erica recommended I not interfere with your day-to-day activities, so you're in charge. You have whatever flexibility you want."

Ann was excited to be working off-campus, away from Jane, among other things. Betty and Carla were as excited about the project as Ann; morale was very high. They came early and stayed late; it felt great to have the chance to develop a really super system.

After a week, Ann met with Jane. "We have some great ideas to do what the students want; it will blow them away!" Jane was pleased Ann was so excited, and didn't want to quash her enthusiasm by asking about the schedule. Ann reported every two weeks that morale was superb and everyone was working very long hours six days a week. Jane asked about schedule, but Erica said not to push too hard for answers so that morale wouldn't be affected.

After 2 ½ months Jane couldn't wait any more and asked, "How are you doing on the schedule?" "We're doing great," Ann said. "We're working hard and it's coming together." "Are you going to be done in two weeks?" "No, no way. It's complicated; it will take more like two months. But it will really spark your cookies."

"Whoa. We have subpoenas from RIAA already and more to come." "Cool down, Jane; the students will love this software." "Ann, we need it now, not in two months. This is a disaster; the administration will be totally pissed. The lawyers have been telling the RIAA to hold off because we're taking care of the problem. They'll look really bad. They'll get skewered at the next meeting with RIAA."

"I'm sorry, Jane. I thought the idea was to build a dynamite system that the students would all use instead of file sharing. I didn't realize schedule was the main objective. And, you know, if we built anything less than what we're doing, the students wouldn't use it. But you're the boss. I'll talk with Betty and Carla and see what we can do."

The developers decided that, with all the design complete and much of the implementation, the fastest way to finish would be to continue as planned. They continued to work very hard but their estimates were poor. After two months, Jane moved them back on campus, morale nosedived and they quit working late. They finished after 7 months, 4 months late.

a. List at least 4 errors made by the management and development team in this project. (5 points)

1. Lack of design visibility or design review. They did have requirements, though we don't know how good they were.
2. No software development schedule or project management. No task list.
3. No use of any form of spiral model. Single release at end of development; no intermediate integration (at least none reported or visible).
4. Reliance on individual heroics and overtime, which is very risky. No estimation of function points, lines of code, schedule or person months.
5. Poor communication between developers and manager.

b. What software development method did they use? Is this an appropriate software development method in this case? Why or why not? (5 points)

Skunkworks is a very unstructured development method, not appropriate for projects like this with known requirements that must be completed on a tight schedule. Skunkworks are fine for exploratory projects without tight deadlines, or ones without firm requirements ("do something great for the next release; it doesn't matter what it is, just as long as it gets our customers excited")

c. At the 2 1/2 month crisis point, what software development method changes would you recommend? Justify your choice. (5 points)

1. Design review, and trimming any features that could be made optional.
2. Resource and schedule estimation, based on function points, after the design review, applying appropriate ranges to assess schedule risk.
3. Creation of mini-milestones (tasks either 0% or 100% complete) and tracking them. While this is not a case a project failure as is often the reason for mini-milestones, there is a major risk of progress on larger tasks not being assessed accurately and, with so little time left, it would be a problem.
4. Inform the customer of the status, range of expected completion times, major issues. Attempt to get agreement on schedule vs function tradeoff. Resources are fixed; you can't add staff in a helpful way at this point (other than perhaps a QA person to test).
5. Plan a second cycle of the spiral model after the first release, to get both the developers and customer to agree to move features to it, versus feeling desired features may be dropped forever.

d. What were the key risks of this project, and how would you have managed them throughout the project? (5 points)

1. Lack of visibility of skunkworks effort. Manage by choosing a different software process with more visibility.
2. Lack of bringing the product to a shippable state at any intermediate time. In a 3 month project, it's hard, but a 2 month shippable version should be in the schedule.
3. Having developers offsite. Should have set up, with developers' agreement, a communication or meeting method to provide better visibility.
4. Lack of requirements review and design review. Hold them early in the project.

2. Data model (20 points)

MIT's music project LAMP contains information about songs, bands, albums and artists. Draw a data model (entity-relationship diagram) implementing the following rules.

a. A band can produce many albums.
b. Each album is produced by one band
c. Each album has many songs
d. Each song may be on one or more albums
e. Each artist may be part of many bands
f. Each band consists of one or more artists
g. A band may perform more than one version of a song, but each version must appear on a different album

Albums have a name, date (year), number of songs, length, price, and record company
Bands have a name and agent
Songs have a name, version, one or more writers (artists) and length
Artists have a name, age, gender, and citizenship

Define and use domain entities where appropriate.

You must draw a data model that corresponds to this set of business rules. Follow these steps. You only need to turn in one drawing that includes all the elements listed in steps a-e.

a. Draw a box for each entity: give each an appropriate name. If there are many-to-many relationships in your model, show the associative (intermediate) entity explicitly (5 points)

b. List the attributes in the box for each entity (5 points)

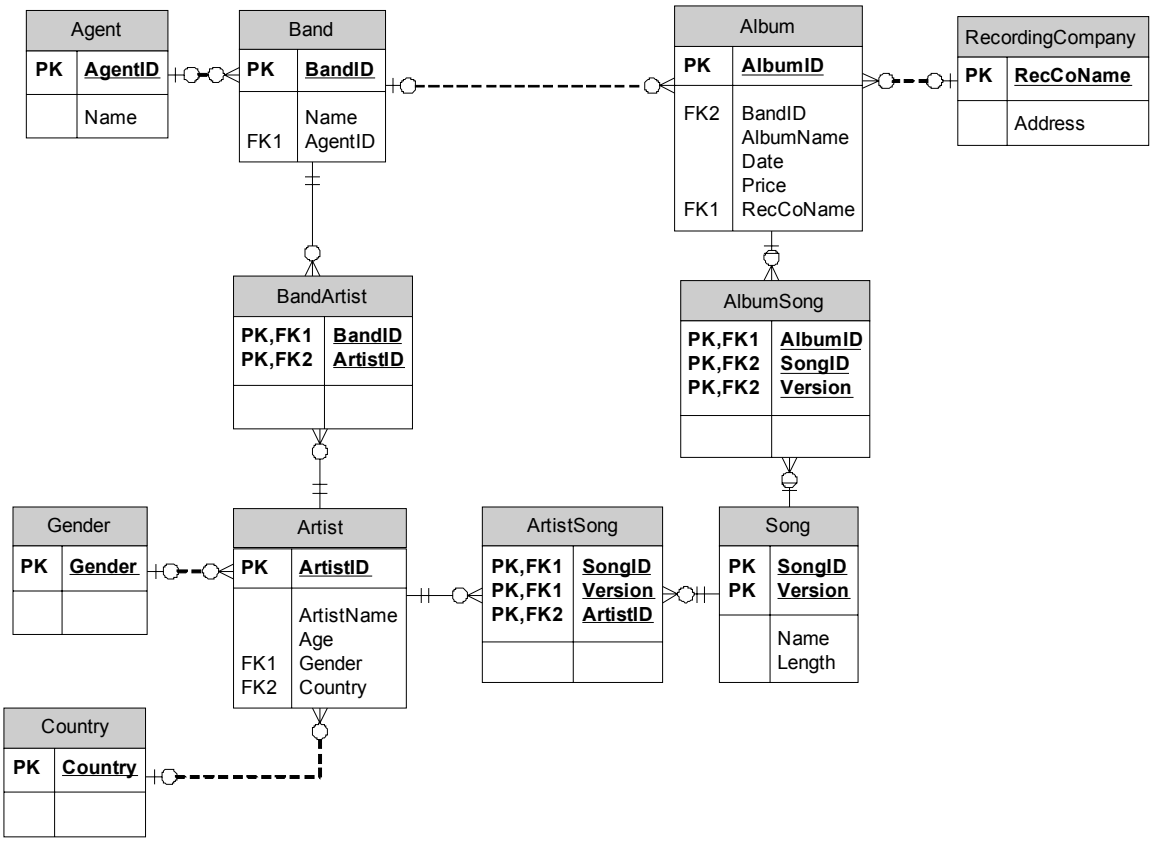
c. **Indicate the primary key for each entity by placing the phrase (PK) next to its name. Ensure that the primary key is unique. (2 points)**

d. **Draw all relationships between the entities in the model. Indicate foreign keys by placing the phrase (FK) next to attributes that are foreign keys. (5 points)**

e. **Indicate the cardinality of the relationship: many-many, many-one or one-one. Use crows-foot notation; if you use another notation, define it. (3 points)**

Normalize these tables; in particular, there must be no duplicated data. Do not store any computed quantities in the tables.

Draw your data model on this page:



Note there is some judgment. It's hard to guarantee uniqueness of album name, artist name, or band name, so IDs are created for them. Record company name should be unique. The 4 domain entities may vary: The Gender entity would not be used by every designer. Agent would be used if there are few agents compared to the number of bands; if each band has a different agent, this entity is not useful. Same reasoning for RecordCompany. Country would generally be present.

Relationships between the 4 major entities (band, song, artist, album) are many-many except for band and album.

3. Database. (20 points)

A database is defined as the following set of tables; the primary keys are bold in each table. The three tables are Courses, Grades and Students. Example data is shown below.

CourseID	Dept	CourseNum
1	CE	1.264
2	ME	2.204
3	AE	3.112

StudentID	**CourseID**	Grade
1	1	A
1	2	A
2	1	B
2	3	C
3	1	A
3	2	C
3	3	F
4	2	A

Name	**StudentID**	Age	GPA
Ann	1	21	4.5
Bob	2	20	3.4
Charles	3	19	4.1
Derya	4	22	5.0

Write the following queries in MS Access SQL or standard SQL syntax. Indicate which kind of SQL syntax you are using and also define any variable you are using which is not obvious. Write queries to get answers to the following questions.

a. Find the students (show their names) who are either younger than 20 or are older than 21. (5 points)

```
SELECT Name
FROM Students
WHERE Age<20 OR Age>21;
```

b. Find the students (show their names) who have taken course 1.264. (5 points)

```
SELECT Students.Name
FROM Students INNER JOIN (Courses INNER JOIN Grades ON Courses.CourseID = Grades.CourseID) ON Students.StudentID = Grades.StudentID
WHERE (((Courses.CourseNum)='1.264'));
```

```
SELECT Students.Name
FROM Students, Courses, Grades
WHERE Courses.CourseID= Grades.CourseID AND Students.StudentID= Grades.StudentID AND Courses.CourseNum= '1.264';
```

c. Find the students (show their names) who have taken a CE or ME course. (5 points)

```
SELECT DISTINCT Students.Name
FROM Students INNER JOIN (Courses INNER JOIN Grades ON Courses.CourseID = Grades.CourseID) ON Students.StudentID = Grades.StudentID
WHERE (((Courses.Dept) In ('CE','ME')));
```

```
SELECT DISTINCT Students.Name
FROM Students, Courses, Grades
WHERE Courses.CourseID = Grades.CourseID AND Students.StudentID = Grades.StudentID AND Courses.Dept In ('CE','ME');
```

d. Find the students (show their names) who have received an A in every class they have taken. (5 points)

Write this as two queries: Call the first one Query3

```
SELECT StudentID, COUNT(*) AS acount
FROM Grades
WHERE Grade='A'
GROUP BY StudentID;
```

SELECT Students.Name
FROM Students INNER JOIN Query3 ON
Students.StudentID=Query3.StudentID
WHERE Query3.acount = (SELECT COUNT(*) FROM Grades WHERE
Grades.StudentID = Query3.StudentID);

The standard SQL queries are virtually identical; change the JOIN syntax in the second query.

You may use more than one SQL statement for part d if you wish.

4. **Web security and Encryption (20 points)**

Public Key Infrastructure (PKI) Encryption uses a public key and a private key to secure messages sent over the web, but there are several ways to execute PKI over the web.

a. **Explain the fundamental difference between asymmetric key infrastructure and symmetric key infrastructure. (5 points)**

Asymmetric: Two keys, a private and public one, are used. All users know the public key; the private key is known only to its owner.

Symmetric: One key is used and must be kept secret to only the two participants in the secure connection. It must be sent across the network securely to the other end of the connection to start the encrypted session.

b. **Describe how one employee Bob, might send a message to Alice, another employee at OmniCron Corporation using asymmetric key encryption. (5 points)**

Bob looks up Alice's public key or requests Alice to send it to him, encrypts his message to her using her public key and sends it. Alice receives the encrypted message, decrypts it with her private key, and reads it.

c. **Clearly explain two benefits of using asymmetric key encryption, instead of using symmetric key encryption, for encryption and/or authentication. (5 points)**

Encryption: Asymmetric private keys do not have to be shared or transmitted across the network. The private key is generated on a computer and never leaves. The public key can be sent to anyone. A symmetric key must be sent securely to the other party.

Authentication. A sender can sign her message with her private key. The receiver decrypts with her public key. If the message decrypts correctly (the receiver should have sent a challenge phrase for the sender to send back), then the receiver knows the sender is who she claims to be.

General benefit. Key management is much simpler with asymmetric keys; they can be published and used by all to send secure messages to recipients with public keys (certificates).

d. **Explain the role of the certificate authority in authentication of a user. (5 points)**

The CA signs the certificates that include the public key of all users. Your browser or client has the CA certificate, with the CA public key. The other party's certificate is signed with the CA private key. You decrypt their certificate with the CA public key and, if it matches (email address, name, URL, depending on certificate type), you know the identity (as defined by email, URL, etc.) of the other party. A chain of CAs can be used.

5. Telecommunications networks and computer hardware. (20 points)

MIT has decided to provide all students with laptop computers and books for all courses. It has placed RFID tags on all laptops and books, and has placed readers at the doorways to all classrooms. There are approximately 10,000 students at MIT. Each student takes an average of 4 classes, which meet 3 times per week on average. Each student has one laptop computer and 8 books, on average. Assume all classes meet on the hour from 9am to 4pm on Monday, Wednesday and Friday only.

a. If the RFID tag transmits 100 bytes of information, what is the total bandwidth (bits/second) required to transmit all RFID data to a central MIT server? Assume all students exit the classroom in a one minute period for each class. (5 points)

1. 10,000 students x 4 classes= 40,000 classes/week
2. Student has 0-9 RFID tags; assume average of 5, so there are 5 x 40,000 = 200,000 RFID scans/week
3. Total data= 200,000 scans/week x 100 bytes/scan x 8 bits/byte= 16 Mbits
4. Time period= 8 class endings/day x 3 class days/week x 1 minute/ending x 60 seconds/minute= 1440 seconds
5. Bandwidth= data/time period= 16 Mbits/1440 seconds= 11 kbps approx.

b. Would a GPRS mobile data network connecting each RFID reader to the central server have sufficient bandwidth? Assume GPRS can provide 40 kbps. (5 points)

Yes. Bandwidth needed of 11 kbps is about ¼ of bandwidth available of 40 kbps even if all traffic were on a single link.

c. Would a 10Mbps Ethernet LAN be sufficient if all the RFID readers were connected to a single LAN? Assume that all classrooms are within a 2km distance. (5 points)

Yes. This data would occupy only a tiny fraction of the LAN bandwidth: 11 kbps out of 6Mbps, even allowing for a practical Ethernet capacity of 60% of its theoretical throughput.

d. Each RFID scan results in 1 disk read and 4 disk writes on the RFID database (indexes, logs, tables and statistics are updated; ID tags are checked for validity; etc.). Would a single disk operating at 140 I/O (input/output, or read/write) operations per second, be sufficient in the server to record all the RFID activity as students exit classrooms? If not, how many disks would be required, and what RAID configuration would you use: how many disks and what type of RAID? (5 points)

1. Transactions/week= 200,000
2. Time period/week= 1440 seconds
3. Transactions/second= 140/second (approx)
4. I/O operations/second= 140 x 5 reads-writes= 700 I/O ops/second

This requires 5 disks if they operate at 140 I/O operations/second at 100% utilization, which is not practical. Realistically you need 8 disks to operate at about 60% utilization just to store the data. If you want data protection (highly recommended), mirroring is usually the best strategy. This would require 16 disks, in a RAID 1+0 configuration (striping and mirroring). If you add 2 disks to mirror the operating system, a 20 disk system would be reasonable. Thus, we'd use two mirrors, each having a 10-disk stripe set.