Practice Material from a Prior First Midterm Exam

The format of the exam is

- (45-50 points) A database design exercise in which you must read a written description of
  a situation, and design an appropriate database, including an entity-relationship diagram
  and a database design outline.
- (20-30 points) Storage calculation and data transfer time exercises.
- (20-30 points) Multiple choice questions, based primarily on the lecture notes. Small
  amounts of material on storage calculations and/or database design may also be included.

Note that there is only practice material for the database design and storage calculations portions
of the exam. There is no practice material for the multiple choice section.

Q1. Keeping track of print advertising

Your firm places advertising in a wide variety of magazines, newspapers, and trade publications.
Your job is to create a database to keep track of your print ads and the contracts, also called
“buys”, through which you pay publishers for running them.

For each ad, you want to store a description, the date its design was finalized, and the name of
the creative director who “signed off” on its design.

You also need to store information on publishers, each of whom may own more than one
publication. For each publisher, you want to store a name and description, along with the
address, city, state, zip code, and phone number of the publisher’s advertising sales department.
Assume you do not have a zip code table available.

For each publication in which you have placed ads (or are considering for future placements),
you want to store a name, description, publication frequency (monthly, weekly, etc.), and current
average circulation. You also need to know which publisher owns it.

A contract, or “buy”, is an agreement with a single publisher. Over time, you will likely make
more than one contract with any given publisher. For each contract, you want to store the date it
was signed and the agreed price. Each contract pays for one or more ad placements. A
placement is an agreement to run a particular ad in a particular issue of a publication. For each
placement, your database should be able to tell you

- Which contract paid for it
- Which of your ads it involves
- Which publication it is in
- Which issue of the publication is involved (which may be stored as a date/time field).

Design a database to hold this information. Draw an entity-relationship diagram and write
a database design outline. You may create “ID” fields as necessary.
Q2: Memory Storage Calculations

Show your work for each problem.

(a) Your PC has a nice new 24-inch widescreen flat panel monitor, whose resolution is $1920 \times 1200$ pixels. If you are currently using 32 bits of color information per pixel, how much storage would be required to make a “screen shot” of your PC screen, without compression? Express your answer in binary-style MB.

(b) The reptile pet owners association is considering adding a video clip posting capability to its website. A member survey suggests that the number of clips that would be posted on the site would be about 30,000. Assume each clip is $300 \times 400$ pixels resolution, with 24 bits per pixel and 30 frames per second, and that clips average 3 minutes long. Disregarding the audio component of each clip, and assuming you store the clips in a compressed format with an average compression factor of 40, how much storage will be needed for the video clip library? Express your result in binary-style GB.

(c) You use a digital voice recorder to record confusing college lectures. The recorder uses CD-quality sound, that is, 44,100 samples per second with 16 bits per sample, but only records one channel. You have found that you can record 7 lectures, each 80 minutes long, before the 256-MB (binary style) memory card in the recorder fills up. What compression factor is the software on the recorder achieving when it stores the audio information?

Now consider the following table:

PRODUCT(ProductID, Name, Description, Price, ShippingPounds, DateIntroduced, MaintenanceAgreementPrice, MaintenanceAgreementTerms, MaintenanceAgreementMonths)

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Access Datatype</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProductID</td>
<td>Text, 12 characters</td>
</tr>
<tr>
<td>Name</td>
<td>Text, 50 characters</td>
</tr>
<tr>
<td>Description</td>
<td>Text, 200 characters</td>
</tr>
<tr>
<td>Price</td>
<td>Currency</td>
</tr>
<tr>
<td>ShippingPounds</td>
<td>Number: “single” format</td>
</tr>
<tr>
<td>DateIntroduced</td>
<td>Date/Time</td>
</tr>
<tr>
<td>MaintenanceAgreementPrice</td>
<td>Currency</td>
</tr>
<tr>
<td>MaintenanceAgreementTerms</td>
<td>Text, 255 characters</td>
</tr>
<tr>
<td>MaintenanceAgreementMonths</td>
<td>Integer (“short” style)</td>
</tr>
</tbody>
</table>

(d) Estimate the binary-style MB required for the PRODUCT table above, assuming there are 10,000 products.

Because you offer maintenance agreements on only 3% of your 10,000 products, now suppose you split the above table into two as follows:
PRODUCT(ProductID, Name, Description, Price, ShippingPounds, DateIntroduced)

MAINTAINABLEPRODUCT(ProductID, MaintenanceAgreementPrice,
                       MaintenanceAgreementTerms,
                       MaintenanceAgreementMonths)

ProductID foreign key to PRODUCT

(e) With this new design, estimate the total binary-style MB required to store the two tables PRODUCT and MAINTAINABLEPRODUCT. Assume all datatypes are unchanged from part (d).

Solutions

Q1. Keeping Track of Print Advertising

AD(AdID, Description, DateFinalized, SignOffDirector)

PUBLISHER(PublisherID, Name, Description, Address, City, State, Zip, Phone)

PUBLICATION(PublicationID, Name, Description, Frequency, Circulation, PublisherID)

CONTRACT(ContractID, DateSigned, AgreedPrice, PublisherID)

PLACEMENT(PlacementID, ContractID, AdID, PublicationID, IssueDate)

Here, I show a synthetic key for PLACEMENT. If you make the reasonable assumption that you only run a given ad once in a single issue of a publication, (AdID, PublicationID, IssueDate) would be reasonable composite key.
It is tempting to insert an extra relationship between AD and PUBLICATION; that isn’t necessary, because the PLACEMENT table tells you which ads have been placed in each publication.

It is possible to argue, depending on exactly when data gets entered in the various tables, that no direct relationship is needed between CONTRACT and PUBLISHER, since you could figure out this relationship by tracing through foreign keys from CONTRACT to PLACEMENT to PUBLICATION, and then to PUBLISHER. However, since it’s not clear that PLACEMENT entries are made as soon as a contract is created, it is probably safer to keep the CONTRACT-PUBLISHER relationship. In that case, there is actually a clever arrangement of composite keys that guarantees that the publication in each placement has the same publisher as the placement’s contract. I would not expect students to come up with that, however.

There is no need to have an entity for issues of publications. That would only be necessary if we were storing information determined by the combination of PublicationID and IssueDate, such as the number of pages in the issue.

Q2. Memory Storage Calculations

(a) Here, the 24-inch size of the screen is a useless “red herring”. Since we already know the screen’s resolution is 1920 × 1200 pixels, we can just calculate

\[
\left(1920 \times 1200 \text{ pixels}\right) \left(\frac{32 \text{ bits}}{\text{pixel}}\right) \div \left(\frac{8 \text{ bits}}{\text{byte}}\right) = 9,216,000 \text{ bytes} \div \left(\frac{1024^2 \text{ bytes}}{\text{MB}}\right) = 8.79 \text{ MB}
\]

Multiplying by inches is only necessary when the resolution is given in dots per inch. If we already know the total horizontal and vertical pixels, the physical dimensions of the screen are not important.

(b) We calculate:

\[
\begin{align*}
&300 \text{ horizontal pixels} \\
\times &400 \text{ vertical pixels} \\
\times &3 \text{ bytes/pixel} = 24 \text{ bits/pixel} \\
\times &3 \text{ minutes} \\
\times &60 \text{ seconds/minute} \\
\times &30 \text{ frames/second} \\
\times &30,000 \text{ clips} \\
= &58,320,000,000,000 \text{ bytes} \\
\div &40 \text{ compression factor} \\
= &1,458,000,000,000 \text{ bytes} \\
\div &1024^3 \text{ B/GB} \\
= &1358 \text{ GB} (= 1.33 \text{ terabytes})
\end{align*}
\]

(c) We first note that 16 bits/sample = 2 bytes/sample. Then,
\[
\left( \frac{44,100 \text{ samples}}{\text{second}} \right) \left( \frac{2 \text{ bytes}}{\text{sample}} \right) \left( \frac{60 \text{ seconds}}{\text{minute}} \right) \left( \frac{80 \text{ minutes}}{\text{lecture}} \right) (7 \text{ lectures}) = 2,963,520,000 \text{ bytes},
\]

and so we calculate

\[
\text{Compression factor} = \frac{\text{Original size}}{\text{Compressed size}} = \frac{2,963,520,000 \text{ bytes}}{(256 \text{ MB})(1024^2 \text{ bytes/MB})} \approx 11.0.
\]

(d) First, we need to calculate the number of bytes needed for each row of the table:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Access Datatype</th>
<th>Size in Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProductID</td>
<td>Text, 12 characters</td>
<td>12</td>
</tr>
<tr>
<td>Name</td>
<td>Text, 50 characters</td>
<td>50</td>
</tr>
<tr>
<td>Description</td>
<td>Text, 200 characters</td>
<td>200</td>
</tr>
<tr>
<td>Price</td>
<td>Currency</td>
<td>8</td>
</tr>
<tr>
<td>ShippingPounds</td>
<td>Number: “single” format</td>
<td>4</td>
</tr>
<tr>
<td>DateIntroduced</td>
<td>Date/Time</td>
<td>8</td>
</tr>
<tr>
<td>MaintenanceAgreementPrice</td>
<td>Currency</td>
<td>8</td>
</tr>
<tr>
<td>MaintenanceAgreementTerms</td>
<td>Text, 255 characters</td>
<td>255</td>
</tr>
<tr>
<td>MaintenanceAgreementMonths</td>
<td>Integer (“short” style)</td>
<td>2</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>547</strong></td>
</tr>
</tbody>
</table>

We then multiply the 547 bytes per row by 10,000 rows, obtaining

\[
\left( \frac{547 \text{ bytes}}{\text{row}} \right) (10,000 \text{ rows}) = 5,470,000 \text{ bytes} \div \left(1024^2 \text{ bytes/MB}\right) = 5.22 \text{ MB}.
\]

(e) For PRODUCT, we are removing the last three attributes, which occupy 8 + 255 + 2 = 265 bytes/row. Thus, the bytes per row are now 547 – 265 = 282 bytes per row, and since we still have 10,000 rows, we calculate

\[
\left( \frac{282 \text{ bytes}}{\text{row}} \right) (10,000 \text{ rows}) = 2,820,000 \text{ bytes} \div \left(1024^2 \text{ bytes/MB}\right) = 2.69 \text{ MB}.
\]

MAINTAINABLEPRODUCT has the 12-byte field ProductID, plus the 265 bytes for the maintenance agreement fields, for a total of 12 + 265 = 277 bytes/row. Since only 3\% of products have maintenance agreements, we have only (3\%)(10,000 rows) = 300 rows. We then calculate

\[
\left( \frac{277 \text{ bytes}}{\text{row}} \right) (300 \text{ rows}) = 83,100 \text{ bytes} \div \left(1024^2 \text{ bytes/MB}\right) = 0.08 \text{ MB}.
\]

For the total storage for the two tables, we thus obtain 2.69 MB + 0.08 MB = 2.77 MB.