

Rutgers University, Business School/Undergraduate New Brunswick
Operations Management (33:623:370)
Fall 2008; Instructor: Jonathan Eckstein

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. Q1 and Q2 below are examples of such exercises (but the test itself will only have one such question)
- (20-25 points) Storage calculation and data transfer time exercises; Q3 below is an example.
- (25-30 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. There is no practice material here 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. The Metropolitan Zoo

The Metropolitan Zoo has placed you in charge of animal husbandry, which involves both care of individual animals, and cleaning and maintenance of their enclosures. You have decided to move from a paper record-keeping system to a more modern electronic system.

The zoo has over 3,000 animals, representing over 400 species. For each species of animal, you want to store the common name (for example, “Grizzly Bear”), scientific name (for example, *Ursus Arctos Horribilis*), and conservation status (for example, “threatened”). Each individual animal in the zoo has a unique tag number, typically engraved on a small leg band or bracelet. For each animal in the zoo, you also want to store its species, date acquired, birth date (if known), and gender. Most individual animals have names (such as “Corky” or “Scooter”), and the system should also store that information.

You keep the animals in enclosures, most of which contain more than one animal. Each animal is restricted to a single enclosure (to simplify this exercise, assume that you never move animals from one enclosure to another). Each enclosure has a unique “pen number”. For each enclosure, you also want to store its area in square feet, and a description of its location. The system should know which animals are in which enclosures.

Your employees perform periodic health checkups on the animals and maintenance inspections of the enclosures. For each employee, you want to store a first name, last name, date hired, and mobile phone number. Each health checkup involves a single employee and single animal. In addition to this information, you want to store the date and time of the checkup, and any comments the employee wishes to make as a result of the checkup. Each maintenance inspection involves a single employee and a single enclosure. In addition to this information, you want to store the date and time of the inspection, along with any comments the employee wishes to make. Only about 4% of health checkups and 5% of maintenance inspections result in any comments.

Design a database to hold all this information. Draw an entity-relationship diagram and write a database design outline. You may create “ID” fields as necessary. If possible, create a design that avoids wasting space on empty comment fields.

Q3. 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 1920×1200 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 300×400 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)

<u>Attribute</u>	<u>Access Datatype</u>
ProductID	Text, 12 characters
Name	Text, 50 characters
Description	Text, 200 characters
Price	Currency
ShippingPounds	Number: "single" format
DateIntroduced	Date/Time
MaintenanceAgreementPrice	Currency
MaintenanceAgreementTerms	Text, 255 characters
MaintenanceAgreementMonths	Integer ("short" style)

- (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)

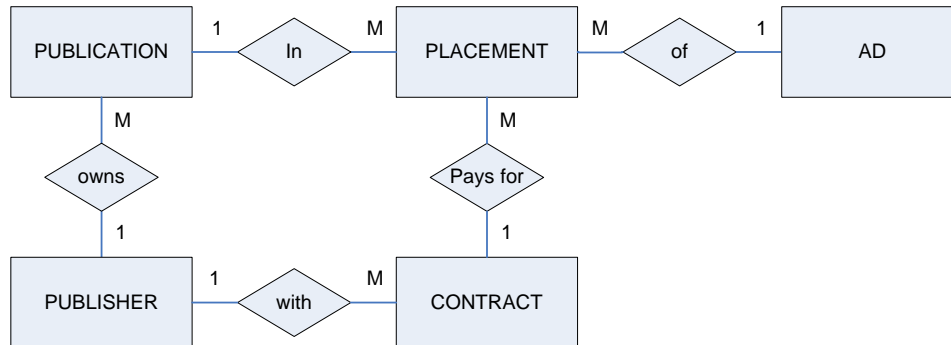
MAINTAINABLEPRODUCT(ProductID, MaintenanceAgreementPrice, MaintenanceAgreementTerms, MaintenanceAgreementMonths)

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)
PublisherID foreign key to PUBLISHER

CONTRACT(ContractID, DateSigned, AgreedPrice, PublisherID)
PublisherID foreign key to PUBLISHER

PLACEMENT(PlacementID, ContractID, AdID, PublicationID, IssueDate)
ContractID foreign key to CONTRACT
AdID foreign key to AD
PublicationID foreign key to PUBLICATION

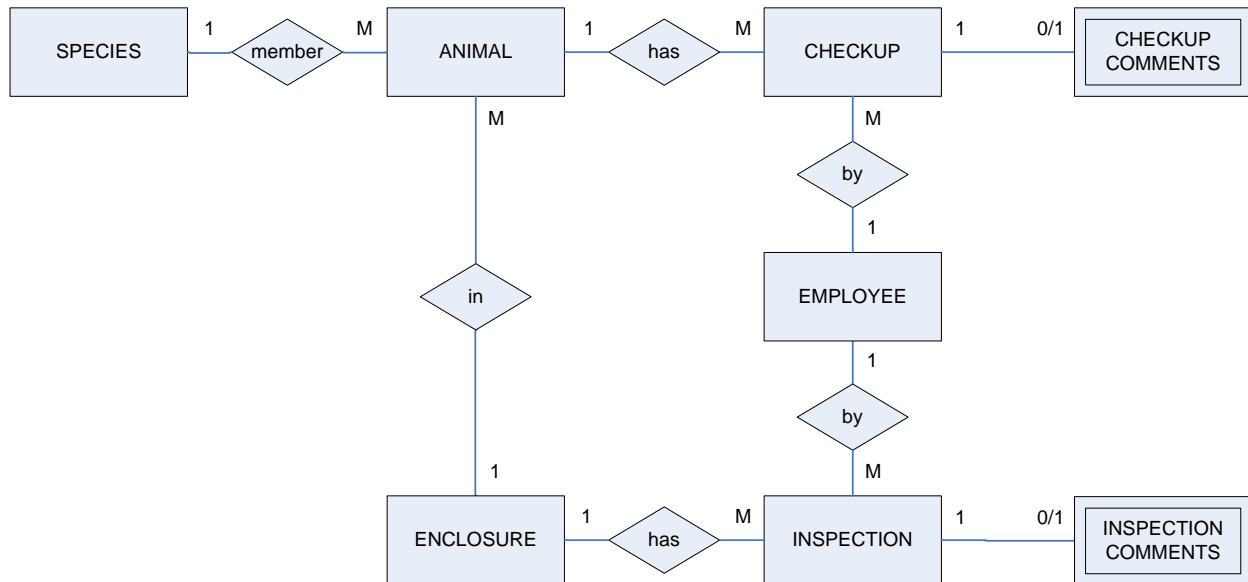
Here, I show a synthetic key for PLACEMENT. If you make the reasonable assumption that you only run a given ad once in a single issue of a publication, (*AdID*, *PublicationID*, *IssueDate*) would be reasonable composite key.

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 the 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. The Metropolitan Zoo



SPECIES(SpeciesID, CommonName, ScientificName, ConservationStatus)

ANIMAL(TagNumber, Name, DateAcquired, BirthDate, Gender, SpeciesID, PenNumber)
 SpeciesID foreign key to SPECIES
 PenNumber foreign key to ENCLOSURE

ENCLOSURE(PenNumber, Area, LocationDescription)

EMPLOYEE(EmployeeID, FirstName, LastName, DateHired, MobilePhone)

CHECKUP(CheckupID, Employee, TagNumber, DateAndTime)
 EmployeeID foreign key to EMPLOYEE
 TagNumber foreign key to ANIMAL

CHECKUPCOMMENT(CheckupID, Comment)
 CheckupID foreign key to CHECKUP

INSPECTION(InspectionID, Employee, PenNumber, DateAndTime)
 EmployeeID foreign key to EMPLOYEE
 PenNumber foreign key to ENCLOSURE

INSPECTIONCOMMENT(InspectionID, Comment)
 InspectionID foreign key to INSPECTION

For SPECIES, I have shown a synthetic key, which I think is the simplest option. The scientific name ought to be unique, and thus an acceptable (if somewhat long) choice for the primary key. The common name would be riskier, and not a wise choice of primary key.

In ANIMAL, Name is blank if an individual animal has no name, and BirthDate is blank if the birth date is unknown. I did not specify what fraction of animals have names or known birth dates, but it probably would not be worth creating subtypes to avoid blank fields in ANIMAL.

With regard to the subtypes for checkup and inspection comments, a simpler approach would be to delete them, and just put Comment fields into CHECKUP and INSPECTION, and delete the tables CHECKUPCOMMENT and INSPECTIONCOMMENT. However, since Comment should be a long text field, it will usually be blank, and there will be large numbers of checkups and inspections over the years, this simple approach will waste a lot of storage.

I have shown synthetic primary keys for CHECKUP and INSPECTION, which is the simplest approach. Under various reasonable assumptions, a variety of composite keys are also possible: *(Employee, DateAndTime)* or *(TagNumber, DateAndTime)* for CHECKUP, and *(Employee, DateAndTime)* or *(PenNumber, DateAndTime)* for INSPECTION. In such cases, the primary keys for the respective subtypes CHECKUPCOMMENT and INSPECTIONCOMMENT should be chosen to match. Note that *(Employee, TagNumber)* is not a possible primary key for CHECKUP, since it would mean that the same employee could not check the same animal more than once. Similarly, *(Employee, PenNumber)* is not a possibly primary key for INSPECTION.

Q3. 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

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

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:

$$\begin{array}{r} 300 \text{ horizontal pixels} \\ \times 400 \text{ vertical pixels} \\ \times 3 \text{ bytes/pixel} = 24 \text{ bits/pixel} \\ \times 3 \text{ minutes} \\ \times 60 \text{ seconds/minute} \\ \times 30 \text{ frames/second} \\ \times \underline{30,000 \text{ clips}} \\ = 58,320,000,000,000 \text{ bytes} \end{array}$$

$$\begin{aligned}
& \div \frac{40 \text{ compression factor}}{1,458,000,000,000 \text{ bytes}} \\
& = \frac{1,458,000,000,000 \text{ bytes}}{1024^3 \text{ B/GB}} \\
& = \boxed{1358 \text{ GB (= 1.33 terabytes)}}
\end{aligned}$$

(c) We first note that 16 bits/sample = 2 bytes/sample. Then,

$$\left(44,100 \frac{\text{samples}}{\text{second}}\right) \left(2 \frac{\text{bytes}}{\text{sample}}\right) \left(60 \frac{\text{seconds}}{\text{minute}}\right) \left(80 \frac{\text{minutes}}{\text{lecture}}\right) (7 \text{ lectures}) = 2,963,520,000 \text{ bytes},$$

and so we calculate

$$\text{Compression factor} = \frac{\text{Original size}}{\text{Compressed size}} = \frac{2,963,520,000 \text{ bytes}}{(256 \text{ MB}) \left(1024^2 \frac{\text{bytes}}{\text{MB}}\right)} \approx \boxed{11.0}.$$

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

Attribute	Access Datatype	Size in Bytes
ProductID	Text, 12 characters	12
Name	Text, 50 characters	50
Description	Text, 200 characters	200
Price	Currency	8
ShippingPounds	Number: "single" format	4
DateIntroduced	Date/Time	8
MaintenanceAgreementPrice	Currency	8
MaintenanceAgreementTerms	Text, 255 characters	255
MaintenanceAgreementMonths	Integer ("short" style)	2
Total		<u>547</u>

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

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

(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

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

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

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

For the total storage for the two tables, we thus obtain $2.69 \text{ MB} + 0.08 \text{ MB} = \boxed{2.77 \text{ MB}}$.