Practice Material for Final Exam

The final will be a three-hour cumulative exam. The planned format is:

- A short section on file sizes and data transfer times, with practice material in Section 1 below; you may also review the “Memory Storage Calculations” handout, the first midterm, question 3 of homework assignment 2, and page 19 of the class notes.
- Two questions on database design and one question on database normalization. Practice material may be found in Sections 2a and 2b below, respectively; you may also review both midterm exams, your class notes, and most of the homework assignments.
- A section on queries similar to the second midterm, with one database and four or five queries, with practice material in Section 3 below. You may also review the second midterm and the associated practice material.
- A multiple-choice section covering all the lecture material. Prepare for this section by studying both sets of lecture notes distributed in class (or a single combined set you may download from the class website). Time permitting, you could also review RT chapters 1-5, 10, TG.4, and TG.5, but studying the other materials mentioned here will likely be a more efficient use of your time. There is no multiple-choice practice material available. However, you should anticipate the questions to be fairly in-depth – many will be difficult to answer without careful study of the lecture notes.

1. Memory Calculations

You have produced a 3-minute marketing video presentation describing a new, fuel-efficient car. The video has 400 × 600 resolution, 25 frames per second, and 24-bit color. For the purposes of this question, you may ignore the audio component of the presentation.

(a) Without compression, how many binary-style GB of storage will the video occupy?

(b) How many binary-style MB will the video occupy if stored via a compression algorithm that has an average compression factor of 20?

(c) A customer connects to the internet via a DSL line that downloads at 768 Kb/s. Estimate how many minutes it will take the customer to download the compressed version of the video.
2a. Database Design Practice Material

2a.i: Rent-a-Wreck

You are setting up a database for a local “rent-a-wreck” car-rental firm. For each vehicle in the firm’s fleet, you want to store the license plate number, manufacturer, model, model year, date acquired, mileage when acquired, and special notes. Assume that all your vehicles are registered in the same state. Each vehicle is assigned to exactly one vehicle type, such as “economy”, “compact”, “intermediate”, “convertible”, “SUV”, and so forth. Each vehicle type has a description, a daily rental rate, and a weekly rental rate.

For each of your customers, you want to store a driver’s license number and state, first name, middle name/initial, last name, date of birth, street address, city, zip code, phone number, and alternate phone number. Assume that your insurance only allows you to rent to U.S. customers, and you do not have access to a zip code table.

Customers place reservations by specifying the type of vehicle they want, the date and time they wish to pick it up, and the date and time they intend to return it.

Vehicles may be rented both as result of reservations, and as the result of “walk-ins”. When a vehicle is rented, you need to store which exact vehicle was rented, and by which customer. For each rental, you want to store the date and time the vehicle was rented, the date and time the customer promised to return it, and the mileage and fuel tank level at the time of rental. When a vehicle is returned, you also want to store the actual date and time of return, mileage upon return, fuel tank level upon return, and special notes.

Design a database to store all this information. Draw an entity-relationship diagram and write a database design outline. You may create “ID” fields as necessary.

2a.ii. Plumbing Parts

You have been assigned to design a parts database for a company that manufacturers plumbing fixtures. Each part is identified by an 8-character part number, and many parts are built out of pieces, which are parts themselves. For example:

| Part #:   | 4MC73498     |
| Description: | Shower valve mixer cartridge |
| Inventory: | 2,340 units currently in stock |
| Contains:  | 1 plastic mixer shell, part # 4MS87349 |
|           | 3 rubber O-gaskets, 12 millimeter diameter, part # 1OG3M12F |
|           | 2 balancing springs, part # 3BX34980 |

The database needs to keep track of the kinds of information shown above, including what parts are contained in what other parts. Furthermore, some parts have special properties:

- Retail: some parts (mostly complete faucet assemblies and the like) are sold as retail products. In this case, there is additional information associated with the part, such as wholesale price, suggested retail price, and shipping weight.
• **Replacement:** Some parts are sold as replacement parts through a nationwide toll-free service number. These parts have a replacement part price and a shipping weight.

Note that any particular part may be a retail part, a replacement part, both, or neither. If a part is both a retail and replacement part, its replacement part price may differ from its wholesale and retail price, and (because of different packing for shipment) its replacement part shipping weight may differ from its retail shipping weight.

Some parts may be obtained from outside suppliers, and in some cases more than one external supplier. Conversely, some outside suppliers may supply more than one part. Each supplier has an ID, a name, a phone number, and address information; assume all suppliers are domestic, and you do not have a zip code table. You order parts from external suppliers in “lots”: for example, a particular supplier only ships rubber O-gaskets in packages of 500. The lot size and price per lot may vary by supplier; for example another supplier might only ship rubber O-gaskets in packages of 1000, and charge a different price per lot.

**Design a database to store all this information. When possible, avoid having to store fields containing blank, “null”, or “does not apply” values. Draw an entity-relationship diagram and write a database design outline. You may create additional “ID” fields as necessary.**

**2a.iii. Custom Cabinetry Shop**

Bill Carpenter is creating a database to keep price list and ordering records for Wood-Maid Cabinetry, located in central Pennsylvania. The firm makes high-quality cabinets, mostly for kitchen renovations. Wood-Maid currently markets 65 different basic kinds of cabinet, each known by a 4- to 8-character code; for example, an “F24D4” is a 24-inch wide, four-drawer floor-standing cabinet. Each kind of cabinet has a base price, a description, and some additional information such as its shipping weight, the type of carton needed for shipping, the number of labor hours needed to build it, and the number of knobs or handles it needs. To fully specify a cabinet, however, the customer must also indicate:

- The kind of wood: Wood-Maid offers 6 different kinds of wood, each with a name and corresponding two-letter code (MA=maple, PI=pine, etc.). For each kind of wood, there is a percentage markup that is applied to the base cabinet price.
- The kind of finish applied to the wood: there are currently 15 different finishes, each with a name, a 3-letter code, and a percentage markup (again, applied to the base cabinet price). For example, a “natural” finish has the code “NAT” and has a markup of 0%. A “pickle” finish has the code “PIC” and a 5% markup.
- Door style: there are 31 different door styles, each with a name, a 4-letter code, a description, and a percentage markup.
- Hardware: “hardware” refers to knobs and handles, which Wood-Maid obtains from a single supplier. There are 52 different styles, each identified by the supplier’s 5-character catalog code. Each style has a description and a price per knob/handle.

Wood-Maid’s customers consist mostly of home improvement contractors. For each customer, Wood-Maid wants to store a company name, a contact first name and last name, a phone num-
ber, and an address, city, state, and zip (assume you do not have access to a zip code table).
Each customer has a main Wood-Maid sales/design representative, who assists in kitchen design
and choosing cabinetry. Each customer is also assigned to a backup representative, in case the
main representative is not available. The main representative for one customer might be the
backup representative for another customer. For each representative, we want to store a first
name, last name, date hired, and telephone extension.

Customers place orders, which typically contain multiple cabinets: for example, three 24-inch 4-
drawer floor cabinets, one 36-inch roll-out floor cabinet, and four 24-inch standard wall cabi-
nets. To avoid costly confusion and mistakes, Wood-Maid requires that all the cabinets in a sin-
gle order have the same wood, finish, door style, and hardware choice. For each order, we also
want to store the date placed, delivery request date, and date shipped (for orders which are com-
plete).

Design a database to store all this information. Your design should automatically enforce
the constraint that all cabinets in an order have the same wood, finish, door style, and
hardware. Draw an entity-relationship diagram and write a database design outline. You
may create “ID” fields as necessary.

2a.iv. Law Firm

The Dewey, Cheatham, and Howe (DCH) law firm has grown significantly over the past decade,
and is looking to move to an automated system for keeping billing records. For each case, the
kind of information the firm wants to store looks like the figure shown on the next page.

Each attorney working at the firm has a unique 5-digit employee number and an hourly billing
rate. There are many cases; every case is associated with a single client, but some clients have
more than one case. When a client has several cases, those cases may have a different client
contact names. Each case has a single “responsible attorney”, who is your staff attorney respon-
sible for coordinating all legal work on the case. As shown above, however, other attorneys on
your staff can bill time to the case (for example, because of their special expertise in certain legal
topic areas).

For most clients, DCH stores only the kind of information shown above: name, phone, and ad-
dress information (city, state and zip code; assume you do not have a zip code table). However,
about 5% of clients are pro bono clients for which DCH performs free legal work – for example,
a non-profit organization that finds apartments for homeless families. Such clients are referred
to DCH by the local legal aid society. For such clients, DCH also stores the date of referral by
legal aid, the name and phone number of the legal aid case worker for the client, and the client
ID number in the legal aid society’s client database.

Attorneys who have not reached “partner” level within the firm have a supervisor, who is an-
other attorney within the firm. You want the database to store information on these relation-
ships, so it can be used for queries like “list all cases for which the responsible attorney is super-
vised by Ed Grossberg”.
Design a database to store all this information. If possible, avoid having to store fields containing blank, “null”, or “does not apply” values. You may create additional “ID” fields where necessary. Draw an entity-relationship diagram and write a database design outline.

2b. Database Normalization Practice Material

2b.i. Courses, Students, and Grades

Consider the following single-table database and illustrative sample data. Assume that each course has only one section and one instructor, and that each student has only one major.

<table>
<thead>
<tr>
<th>StudID</th>
<th>StudName</th>
<th>Major ID</th>
<th>Major Name</th>
<th>Course ID</th>
<th>Course Title</th>
<th>Instructor Name</th>
<th>Instructor Office</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>11232</td>
<td>Bruce Green</td>
<td>623</td>
<td>MSIS</td>
<td>370:03</td>
<td>Information Systems</td>
<td>Sam Shine</td>
<td>Levin-215</td>
<td>F</td>
</tr>
<tr>
<td>11232</td>
<td>Bruce Green</td>
<td>623</td>
<td>MSIS</td>
<td>312:02</td>
<td>Marketing</td>
<td>Steven Parks</td>
<td>Core-315</td>
<td>B</td>
</tr>
<tr>
<td>11452</td>
<td>Ben Jaffleck</td>
<td>010</td>
<td>Accounting</td>
<td>425:03</td>
<td>Corporate Finance</td>
<td>LeeYun</td>
<td>Levin-324</td>
<td>A</td>
</tr>
<tr>
<td>12543</td>
<td>Jane Brown</td>
<td>360</td>
<td>Finance</td>
<td>370:03</td>
<td>Information Systems</td>
<td>Sam Shine</td>
<td>Levin-215</td>
<td>C</td>
</tr>
<tr>
<td>12673</td>
<td>Kate Taylor</td>
<td>010</td>
<td>Accounting</td>
<td>212:02</td>
<td>Micro-economics</td>
<td>Yao Zan</td>
<td>Levin-006</td>
<td>D</td>
</tr>
<tr>
<td>12673</td>
<td>Kate Taylor</td>
<td>010</td>
<td>Accounting</td>
<td>370:03</td>
<td>Information Systems</td>
<td>Sam Shine</td>
<td>Levin-215</td>
<td>B+</td>
</tr>
<tr>
<td>13459</td>
<td>Diane Weld</td>
<td>635</td>
<td>Marketing</td>
<td>421:01</td>
<td>Consumer Behavior</td>
<td>Steven Parks</td>
<td>Core-315</td>
<td>A</td>
</tr>
</tbody>
</table>
Redesign this database so that it is in third normal form, and write a database design outline. You may introduce additional “ID” fields if you feel it is necessary.

2b.ii. Interfunctional Teams

The CEO of a large firm has organized some interfunctional teams to study issues such as adopting an integrated IT platform and improving the process for new product development. To keep track of who is serving on what team, the CEO’s secretary has created the following spreadsheet table (not all rows are shown).

<table>
<thead>
<tr>
<th>Team Name</th>
<th>Date Team Formed</th>
<th>Employee Name</th>
<th>Employee Department Code</th>
<th>Full Department Name</th>
<th>E-mail</th>
<th>Phone Extension</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT Integration</td>
<td>7/8/2004</td>
<td>Bill Coddingford</td>
<td>IS</td>
<td>Information Systems</td>
<td>wcodd</td>
<td>5478</td>
</tr>
<tr>
<td>IT Integration</td>
<td>7/8/2004</td>
<td>Marsha Freidrich</td>
<td>MKT</td>
<td>Marketing</td>
<td>mfred</td>
<td>3499</td>
</tr>
<tr>
<td>IT Integration</td>
<td>7/8/2004</td>
<td>Gayle Rosenberg</td>
<td>OPS</td>
<td>Operations</td>
<td>grosen</td>
<td>0415</td>
</tr>
<tr>
<td>IT Integration</td>
<td>7/8/2004</td>
<td>Narendra Patel</td>
<td>R&amp;D</td>
<td>Research and Development</td>
<td>npatel</td>
<td>7808</td>
</tr>
<tr>
<td>Development Process</td>
<td>2/6/2006</td>
<td>Yu-ching Pan</td>
<td>IS</td>
<td>Information Systems</td>
<td>ycpan</td>
<td>5482</td>
</tr>
<tr>
<td>Development Process</td>
<td>2/6/2006</td>
<td>Narendra Patel</td>
<td>R&amp;D</td>
<td>Research and Development</td>
<td>npatel</td>
<td>7808</td>
</tr>
<tr>
<td>Development Process</td>
<td>2/6/2006</td>
<td>George Smith</td>
<td>MKT</td>
<td>Marketing</td>
<td>gsmith</td>
<td>3507</td>
</tr>
<tr>
<td>Demand Forecasting</td>
<td>3/31/2005</td>
<td>Marsha Freidrich</td>
<td>MKT</td>
<td>Marketing</td>
<td>mfred</td>
<td>3499</td>
</tr>
<tr>
<td>Demand Forecasting</td>
<td>3/31/2005</td>
<td>Gayle Rosenberg</td>
<td>OPS</td>
<td>Operations</td>
<td>grosen</td>
<td>0415</td>
</tr>
</tbody>
</table>

For example, the “IT Integration” team consists of Bill Coddingford, Marsha Friedrich, Gayle Rosenberg, and Narendra Patel. Each employee works for only one department.

Design a relational database to keep this same information in third normal form; draw an entity-relationship diagram and write a database design outline. You may introduce additional “ID” fields as necessary.

2b.iii. Print Advertising Placements

The following one-table database keeps track of your firm’s advertisement placements in various magazines. For example, ad 101, a comical piece showing bears eating pizza, ran in the June Field and Stream, the July Outdoor World, the July 15 Business Week, and again in the August Field and Stream. Note that you also ran another ad, a futuristic piece showing a spaceship, in the July 15 Business Week. You may run the same ad in several different issues of a magazine, or different ads in a single issue, but you never run the same ad more than once in a single issue of a magazine.

PLACED(AdID, AdDescription, MagazineID, MagazineName, IssueDate, PublisherID, PublisherName)
Redesign this database so that it is in third normal form. Draw an entity-relationship diagram, and write a database design outline.

### 3. Query Exercises

Consider the following database, used by a service that manages construction projects:

**VENDOR**

```
VENDOR(VendorID, Name, StreetAddress, City, State, Zip, Phone, VTypeID)
VTypeID foreign key to VENDOR
```

**CONTRACT**

```
CONTRACT(ContractID, VendorID, ProjectID, ScheduleStartDate, ScheduleDoneDate, EstimatedCost, ActualStartDate, ActualDoneDate, ActualCharged, WorkDescription)
VendorID foreign key to VENDOR
ProjectID foreign key to PROJECT
```

**PROJECT**

```
PROJECT(ProjectID, Name, Description, JobSiteAddress, JobSiteCity, JobSiteState, JobSiteZip, ScheduleStartDate, ScheduleEndDate, ActualStartDate, ActualDoneDate, ManagementFee)
```
The “State” and “JobSiteState” fields hold standard two-letter abbreviations like “NJ”, “NY”, and “TX”. Complete the grids on the query grid sheet to implement the following queries.

- If a query does not use the “Total” row, cross it out. Otherwise, each of its cells should contain “Group by”, “Where”, or an aggregation operator such as “Sum” or “Avg”.
- Each query should show only the requested information, with columns in the order requested.
- For each column used in a query, remember to put a check in the “Show” box whenever necessary.

(a) Find all cases in which a vendor described as a “plumber” has charged an amount exceeding their contract estimate by more than 20%. For each such case, show the vendor name, project name, estimate amount, and amount actually charged.

(b) Show the name, total estimated cost, and total actual cost of each project completed on or before April 30, 2006. List in order of estimated cost, starting with the highest and ending with the lowest. The ManagementFee amount in the PROJECT table should not be considered part of the total estimated or actual cost.

(c) For each vendor with description “plumber” or “HVAC”, list their name, vendor type description, total number of contracts, and average actual cost of each contract.

(d) Find all cases where an “electrician” vendor is from a different state than the job site they are working on. Show the vendor name, vendor state, project description, and job site state. List in order of job site state.

(e) For each vendor, show its name and the number of its contracts for job sites in New Jersey that were completed late or over budget. If there are no such contracts, the vendor need not be listed.