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

Practice Material from a Prior Second Midterm Exam

The format of the exam will be as follows (point totals are approximate and subject to change):

- (30 points) Query grid exercises (on paper)
- (40 points) A database design problem
- (30 points) A database normalization problem

The problems below are representative of a single exam, although Q2 might be a bit more complicated than what you might see on the actual exam. Since the last exam, we have covered many-to-many relationships, as well as multiple relationships between the same pair of tables. Between the database design and normalization problems, you should expect to see at least one many-to-many relationship.

Q1. Query grid exercises

Part of the database in the local public library's information system has the following form:

AUTHOR(AuthorID, Name, Nationality, BirthYear, DeathYear)

BOOK(ISBN, Title, YearPublished, NumPages, Edition, Category, Language, Publisher)

WROTE(AuthorID, ISBN, AuthorPosition)
AuthorID foreign key to AUTHOR
ISBN foreign key to BOOK

COPYOFBOOK(BarCode, ISBN, DateAcquired)
ISBN foreign key to BOOK

The library may have more than one copy of the most popular books, each with a different bar code, but the same ISBN. *DeathYear* is blank for authors who are still alive. *YearPublished*, *BirthYear*, and *DeathYear* are integer fields that hold only a year, not a more detailed date. The WROTE table allows the database to handle books having multiple authors, and its *AuthorPosition* attribute indicates the order in which each author appears on the book's title page. For example, if the book with ISBN 00-783-986392 has the authors Karl Studden (AuthorID 59201), Andras Borovik (AuthorID 27133), and William Cooke (AuthorID 30332), listed in that order, then WROTE would contain the records:

AuthorID	ISBN	AuthorPosition
59201	00-783-986392	1
27133	00-783-986392	2
30332	00-783-986392	3

Complete the grids on the query grid sheet to implement the following queries. Notes:
If you use the "Total" row at all, it should contain either "Group by", "Where", or an aggregation operator such as "Sum" or "Avg" for every column in the query. When you don't need the

"Total" row, cross it out. For each column used in a query, check the "Show" box whenever it is necessary. Each query's output should contain only the requested columns, in the order specified.

(a) Show the title, category, and publication year of all Spanish-language books published in or after the year 2002.

(b) Show the title, name of first author, publication year, and number of pages for all books with at least 1000 pages. List them in order of number of pages, longest books first.

(c) Produce a table that shows the title of each book in the library and the number of copies of it the library owns. Books with different ISBN's but the same title should be treated as different books.

(d) Show the title, year of publication, and number of pages for all books whose category is "fiction" and have more than one author.

(e) Show the names of authors whose nationality is "USA", along with the number of books for which they are the first author and that were published either before 1939 or after 1945, and the average number of pages for such books. Authors who did not write any such books need not appear in the output.

Q2. Database design: the Jersey Pacific Railroad

The Jersey Pacific Railroad (JPRR) needs a database to track the past and present movement of locomotives, freight cars, and freight. (Note: this problem is a simplification of a real railroad.)

A "terminal" is a place where a train may begin or end its journey, and is known by a unique four-letter code. For each terminal, the system should also store a name, description, GPS latitude, GPS longitude, and feet of available track storage space.

Each train ever operated by the JPRR has a unique ID number and travels from a single "origin" terminal to a single "destination" terminal. The system should record the origin and destination of each train, the departure date and time, and the arrival date and time.

Each train carries one or more "shipments". For each shipment, the system should record an ID number, a description of the freight involved (examples: gravel, scrap metal, wheat, or "empty"), a total weight, a total volume, and a negotiated price. Each shipment is also associated with a single customer; for each customer, the system should store a name, billing address, city, state, and zip code, along with a phone number. Assume the JPRR does not have a zip code table. For simplicity, assume that each shipment is part of only one train (and thus has the same origin and destination as that train).

Each shipment uses one or more freight cars. On a given train, a single freight car is never "split" between two shipments; it always carries goods from just one shipment. However, once a train has arrived at its destination, each of its freight cars of course becomes free to be part of a different shipment on a different train. The system should track which freight cars are in each

shipment. For each freight car, the system should store a unique registration number, a type (boxcar, hopper car, tank car, etc.), empty weight in tons, cargo capacity in tons, length in feet, and date built.

Finally, each train is hauled by one or more locomotives. Each locomotive in the JPRR's fleet has a unique "placard" number. In addition to this number, the system should store the locomotive's manufacturer, model, horsepower, weight in tons, and year built. The system should be able to record which locomotives hauled each train. Once a locomotive has finished with one train, it of course becomes free to haul a different train.

Design the database required for the JPRR's system. Draw an entity-relationship diagram and write a database design outline. You may create "ID" fields where necessary.

Q3. Database normalization: the Piscataway Stock Exchange

The table on the last page shows part of the data stored by the Piscataway Stock Exchange (PSE). The information in the first nine rows of table should be interpreted as follows:

- At 9:37 AM on August 23, 2006, MegaBrokers sold 1000 shares of TechMax (ticker symbol TMAX) for $8.24 per share. TechMax is categorized as a small cap (SMC) and information technology (IT) stock.
- One minute later, Kirt Kerorian bought this same block of shares for $8.26 per share.
- At 9:41 AM, MegaBrokers sold another block of shares: 2500 shares of Alstom, SA, which is categorized as large cap (LGC), manufacturing (MAN), and European (EUR). The price was $12.56 per share.
- Also at 9:41 AM, TradePartners sold 700 shares of Whole Foods Market (WFMI) at $20.41 per share. Whole Foods is categorized as a mid cap (MDC) and retail (RET) stock.

The rest of the table should be interpreted similarly. As suggested in the table, you should treat the assignment of stocks to categories as essentially static: for instance, if TechMax is considered a small cap and information technology stock during one transaction, it should be considered a small cap and information technology stock for the purposes of any other transaction.

Design a third-normal-form database to store information like that shown in the table. Draw an entity-relationship diagram and write a database design outline.

Trans ID	*DateAndTime*	*CustID*	*Cust Name*	*Trans Type*	*Ticker Symbol*	*Stock Name*	*Num Shares*	*Unit Price*	*Categ Code*	*Category Descrip*
567823	23-Aug-2006 9:37	789	MegaBrokers	Sell	TMAX	TechMax	1000	$ 8.24	SMC	Small Cap
									IT	Info Tech
567824	23-Aug-2006 9:38	989	Kirt Kerorian	Buy	TMAX	TechMax	1000	$ 8.26	SMC	Small Cap
									IT	Info Tech
567825	23-Aug-2006 9:41	789	MegaBrokers	Sell	ALO	Alstom, SA	2500	$12.56	LGC	Large Cap
									MAN	Manufacturing
									EUR	Europe
567826	23-Aug-2006 9:41	320	TradePartners	Sell	WFMI	Whole Foods	700	$20.41	MDC	Mid Cap
									RET	Retail
567827	23-Aug-2006 9:42	789	MegaBrokers	Buy	WFMI	Whole Foods	500	$20.43	MDC	Mid Cap
									RET	Retail
567828	23-Aug-2006 9:42	989	Kirt Kerorian	Buy	WFMI	Whole Foods	200	$20.43	MDC	Mid Cap
									RET	Retail
567829	23-Aug-2006 9:42	210	ElecTrade	Buy	ALO	Alstom, SA	2500	$12.58	LGC	Large Cap
									MAN	Manufacturing
									EUR	Europe
567830	23-Aug-2006 9:44	320	TradePartners	Sell	MSFT	Microsoft	4000	$43.82	LGC	Large Cap
									IT	Info Tech
567831	23-Aug-2006 9:44	776	Max Selmer	Sell	MSFT	Microsoft	1000	$43.82	LGC	Large Cap
									IT	Info Tech
567832	23-Aug-2006 9:45	789	MegaBrokers	Buy	MSFT	Microsoft	5000	$43.84	LGC	Large Cap
									IT	Info Tech
567833	23-Aug-2006 9:45	989	Kirt Kerorian	Sell	TM	Toyota	800	$51.00	LGC	Large Cap
									MAN	Manufacturing
									AS	Asia
567834	23-Aug-2006 9:46	776	Max Selmer	Buy	TM	Toyota	800	$51.02	LGC	Large Cap
									MAN	Manufacturing
									AS	Asia