Dr. Stefan Brass
School of Information Sciences
University of Pittsburgh

April 26, 2000

INFSCI 2710 "Database Management" — Final Exam —

Instructions

- You have time until 8^{50} to work on the exercises, i.e. 1 hour and 50 minutes.
- Please put the answers into the provided space. If the space is not sufficient, use the back page or attach additional sheets. But mark clearly that there is a continuation, and put your name and the number of the exercise on any additional sheets.
- Please ask if you are not sure that you understood the exercise.
- This exam has 13 pages. Please check that they are complete.
- Please do not take a copy of this exam with you, also do not take notes about the exercises. You will get the exam back when it is corrected.

Please Fill In:

Name: ____________________________________

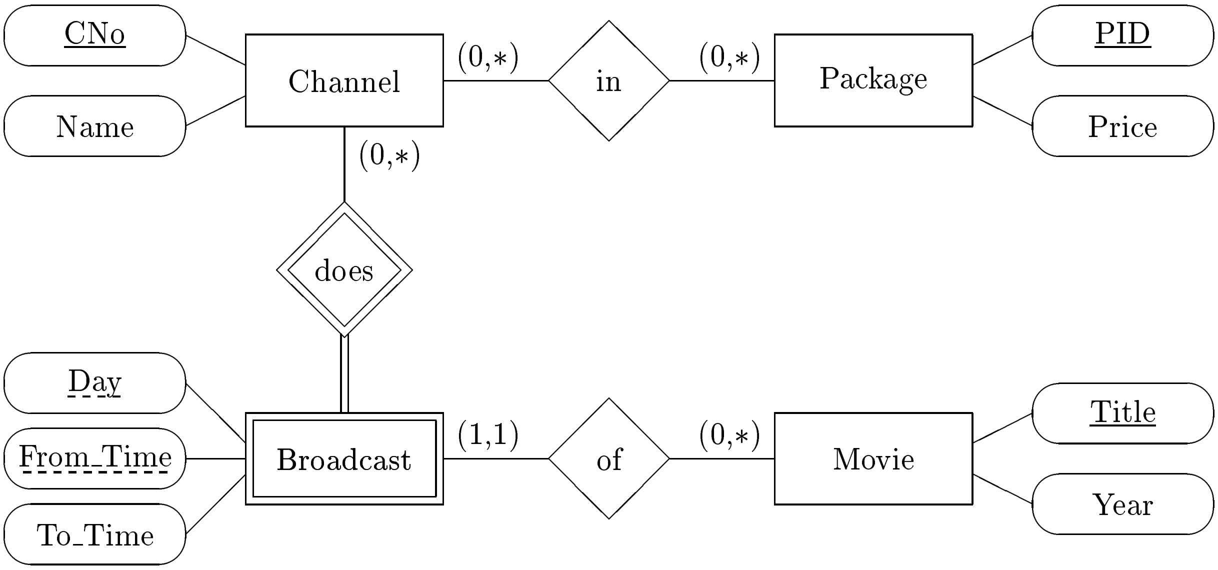
☐ Please check here if this is your last term, so a G (incomplete) grade would not help you.

____ Please tell me the minimum grade which you would find acceptable before you would prefer a G grade and lots of extra work (in order to change your G grade into a real grade). There is no guarantee that I will give you a G grade since I can give only very few students an incomplete grade. Also if this is your last term, please tell me the minimum grade before you would be severely disappointed. You can leave this item blank.

☐ Please check here if you would promise me to do something worth 1% of extra credit (e.g. a small project or 2 pages with suggestions for improving the course) if only this 1% is missing for a better grade. You would not get a G grade in this case. I hope that I will find cutoff points where this will not happen.

Exercise 1 (Translation: ER to Relational Model) 10 Points

Consider a database with information about cable channels like “The Disney Channel”, “ENCORE”, “CINEMAX”, “STARZ!”. We want to store which movies they show at what time (i.e. their program), in which cable packages they are contained, and how much each cable package costs per month.



Please translate this schema into the relational model. You can use an abbreviated syntax for the relations/tables, e.g. $R(A_1, \ldots, A_n)$. But you can also use table skeletons, or, if you prefer that, SQL CREATE TABLE statements. Please specify keys, foreign keys, and optional attributes (if any). If other constraints should be needed to make the relational schema equivalent to the given ER-diagram, it suffices to sketch them in natural language.

Example Database for Exercise 2 and 3

Let us consider a simplified part of the Oracle Data Dictionary which deals with tables and indexes. In this example, all indexes correspond to keys (therefore, they are `UNIQUE`). So if you do not feel comfortable with indexes, you can think of keys.

The table `CAT` contains all tables (and view, synonyms, sequences) which you created:

CAT	
TABLE_NAME	TABLE_TYPE
STUDENTS	TABLE
EXERCISES	TABLE
RESULTS	TABLE

The table `COLS` contains a row for every column in a table or a view (`COLUMN_ID` is the position of that column, e.g. 1 for the leftmost column):

COLS			
TABLE_NAME	COLUMN_NAME	NULLABLE	COLUMN_ID
STUDENTS	STUD_ID	N	1
STUDENTS	FIRST_NAME	N	2
STUDENTS	LAST_NAME	N	3
STUDENTS	EMAIL	Y	4
STUDENTS	PASSWORD	N	5
STUDENTS	LAST_TERM	Y	6
STUDENTS	GRADE	Y	7
EXERCISES	CAT	N	1
EXERCISES	EX_NO	N	2
EXERCISES	TOPIC	N	3
EXERCISES	MAX_POINTS	N	4
RESULTS	CAT	N	1
RESULTS	EX_NO	N	2
RESULTS	STUD_ID	N	3
RESULTS	POINTS	N	4
RESULTS	ENTERED	N	5

The table `IND` contains one row for each index:

IND		
INDEX_NAME	TABLE_NAME	UNIQUENESS
STUDENTS_KEY	STUDENTS	UNIQUE
STUDENTS_NAMES_MUST_BE_UNIQUE	STUDENTS	UNIQUE
EXERCISES_KEY	EXERCISES	UNIQUE
RESULTS_KEY	RESULTS	UNIQUE

Finally, USER_IND_COLUMNS contains one row for each indexed column. For instance, the index STUDENTS_NAMES_MUST_BE_UNIQUE is created on the combination of FIRST_NAME and LAST_NAME. So it has one entry for FIRST_NAME with COLUMN_POSITION = 1 and one entry for LAST_NAME with position 2. You can assume that positions always start with 1 and are consecutively numbered.

USER_IND_COLUMNS			
INDEX_NAME	TABLE_NAME	COLUMN_NAME	COLUMN_POSITION
STUDENTS_KEY	STUDENTS	STUD_ID	1
STUDENTS_NAMES_MUST_BE_UNIQUE	STUDENTS	FIRST_NAME	1
STUDENTS_NAMES_MUST_BE_UNIQUE	STUDENTS	LAST_NAME	2
EXERCISES_KEY	EXERCISES	CAT	1
EXERCISES_KEY	EXERCISES	EX_NO	2
RESULTS_KEY	RESULTS	CAT	1
RESULTS_KEY	RESULTS	EX_NO	2
RESULTS_KEY	RESULTS	STUD_ID	3

Exercise 2 (SQL CREATE TABLE) 5 Points

Write a CREATE TABLE command for "COLS" (it is actually a view, or really a synonym, but for the sake of this exercise let us assume it is a table).

- "TABLE_NAME" is a foreign key referencing "CAT".
- The columns "TABLE_NAME" and "COLUMN_NAME" form together an alternative key.
- None of the columns of "COLS" can be null.
- Table names and column names in Oracle can be up to 128 characters long. Four digits are needed for for the "COLUMN_ID".
- Please use constraints to ensure that "NULLABLE" can be only "Y" or "N" and that "COLUMN_ID" is positive. Please give a name to at least one constraint. You do not have to name the other constraints.

Exercise 3 (SQL) 18 Points

Formulate the following queries in SQL. Your queries must work with any database state, not only with the example. You get three points for every correct query. You may lose points also for unnecessary complications or duplicate output rows. You may define views and use them in the queries. I will accept every SQL construct which works in SQL-92, Oracle, DB2, or SQL Server and which I know. Since I am not yet an expert for DB2 or SQL Server, it is safer if you explain constructs which run only on these systems (or better still, to write portable SQL).

a) Print the names of all indexes which include the two columns “`FIRST_NAME`” and “`LAST_NAME`” of the table “`STUDENTS`”. Only indexes which contain both columns together should be printed. You do not have to check that there are no other columns in that index, and you do not have to ensure specific column positions. In the given example database state, the query result should be:

INDEX_NAME
STUDENTS_NAMES_MUST_BE_UNIQUE

b) Which indexes are only on one column (i.e. which indexes have only one entry in `USER_IND_COLUMNS`)? Print the name of the index, the name of the table, and the column name. In the given example, the query result should be:

INDEX_NAME	TABLE_NAME	COLUMN_NAME
STUDENTS_KEY	STUDENTS	STUD_ID

c) Print for every table, which has at least five columns, the number of columns. The output should contain the table name and this number.

TABLE_NAME	NUM_COLS
STUDENTS	7
RESULTS	5

d) Define a view which contains table name, column name, and ID of all columns which are contained in an index. Note that ID is the column number in the table (from `COLS`), not the position in the index. Also, it is possible that different tables have columns with the same name (of which only one might actually be indexed). There can be overlapping indexes (indexing the same column of the same table). Make sure that every combination of table and column appears only once.

TABLE_NAME	COLUMN_NAME	COLUMN_ID
STUDENTS	STUD_ID	1
STUDENTS	FIRST_NAME	2
STUDENTS	LAST_NAME	3
EXERCISES	CAT	1
EXERCISES	EX_NO	2
RESULTS	CAT	1
RESULTS	EX_NO	2
RESULTS	STUD_ID	3

e) Define a view which contains table name, column name, and ID of all columns which are not contained in any index. Again, ID is the column number in the table, and not the position in the index. Also, different tables can have columns with the same name (as before).

TABLE_NAME	COLUMN_NAME	COLUMN_ID
STUDENTS	EMAIL	4
STUDENTS	PASSWORD	5
STUDENTS	LAST_TERM	6
STUDENTS	GRADE	7
EXERCISES	TOPIC	3
EXERCISES	MAX_POINTS	4
RESULTS	POINTS	4
RESULTS	ENTERED	5

f) Use the views defined in d) and e) in a query which lists all columns (table name, column name, ID) together with an indication whether the column is indexed or not. I.e. you should print "Y" for the columns listed in the view d) and "N" for the columns listed in the view e). Please sort the output by table name and column ID.

TABLE_NAME	COLUMN_NAME	COLUMN_ID	INDEXED
EXERCISES	CAT	1	Y
EXERCISES	EX_NO	2	Y
EXERCISES	TOPIC	3	N
EXERCISES	MAX_POINTS	4	N
RESULTS	CAT	1	Y
RESULTS	EX_NO	2	Y
RESULTS	STUD_ID	3	Y
RESULTS	POINTS	4	N
RESULTS	ENTERED	5	N
STUDENTS	STUD_ID	1	Y
STUDENTS	FIRST_NAME	2	Y
STUDENTS	LAST_NAME	3	Y
STUDENTS	EMAIL	4	N
STUDENTS	PASSWORD	5	N
STUDENTS	LAST_TERM	6	N
STUDENTS	GRADE	7	N

Exercise 4 (FDs, BCNF) 6 Points

Let us consider again the table USER_IND_COLUMNS from the data dictionary. It is actually a view, so normalization is not important for it, but for the sake of this exercise, let us assume that it is a base table.

USER_IND_COLUMNS			
INDEX_NAME	TABLE_NAME	COLUMN_NAME	COLUMN_POSITION
STUDENTS_KEY	STUDENTS	STUD_ID	1
STUDENTS_NAMES_MUST_BE_UNIQUE	STUDENTS	FIRST_NAME	1
STUDENTS_NAMES_MUST_BE_UNIQUE	STUDENTS	LAST_NAME	2
EXERCISES_KEY	EXERCISES	CAT	1
EXERCISES_KEY	EXERCISES	EX_NO	2
RESULTS_KEY	RESULTS	CAT	1
RESULTS_KEY	RESULTS	EX_NO	2
RESULTS_KEY	RESULTS	STUD_ID	3

Assume that the following functional dependencies hold:

- INDEX_NAME, COLUMN_POSITION → COLUMN_NAME
- INDEX_NAME → TABLE_NAME
- INDEX_NAME, COLUMN_NAME → COLUMN_POSITION

Please check exactly one answer for every question. Unless I made a mistake, one and only one answer is correct for every subexercise.

a) Does the functional dependency “COLUMN_NAME → COLUMN_POSITION” hold here? It might hold in this example database state although we assume that it does not hold in general.

- ☐ No, the entries for STUD_ID violate this FD.
- ☐ No, the entries for EX_NO violate this FD.
- ☐ No, the entries for FIRST_NAME and LAST_NAME violate this FD.
- ☐ Yes, this FD is satisfied in the given database state.

b) What does the functional dependency “INDEX_NAME → TABLE_NAME” mean?

- ☐ Every table has only one index.
- ☐ Every index is for a uniquely determined table.
- ☐ No index can have more than one column.

c) Somebody tells you that `INDEX_NAME, COLUMN_POSITION` → `TABLE_NAME` holds, too. What is your reaction?

- ☐ This is impossible. It is already violated in the example state.
- ☐ This is true, but not very interesting, since it is weaker than (implied by) the second given FD.
- ☐ This contradicts the given key. If it is really true, we must determine another key.

d) Given the three functional dependencies, is “`INDEX_NAME, COLUMN_NAME`” an alternative key for the relation?

- ☐ Yes.
- ☐ No.

e) Is the relation in BCNF? You may ignore the third FD for this question, since otherwise the correct answer might depend on your answer for d).

- ☐ Yes.
- ☐ No. The FD “`INDEX_NAME, COLUMN_POSITION` → `COLUMN_NAME`” violates BCNF.
- ☐ No. The FD “`INDEX_NAME` → `TABLE_NAME`” violates BCNF.
- ☐ No. Both FDs violate the BCNF condition.

f) What is one important purpose of normalization?

- ☐ We want to minimize the number of joins in queries.
- ☐ We want to enforce a specific class of constraints (FDs) via key constraints.
- ☐ We want to minimize the number of attributes in each relation.

Exercise 5 (Security) 3 Points

Please check exactly one answer for every question.

a) Ann creates a table `SECRET1` and executes the following command:

```
GRANT SELECT, INSERT ON SECRET1 TO BILL.
```

Sometime later she enters this command: `GRANT DELETE ON SECRET1 TO BILL`.
What rights on `SECRET1` does Bill now have?

- ☐ He got the `DELETE` right, but lost the `SELECT` and `INSERT` rights.
- ☐ This command will result in an error message. You cannot grant `DELETE` without granting `SELECT` in the same command.
- ☐ Bill now has `SELECT`, `INSERT` and `DELETE` rights on this table.

b) Ann creates a table `SECRET2` and executes the following command:

```
GRANT SELECT ON SECRET2 TO BILL WITH GRANT OPTION.
```

Then Bill executes: `GRANT SELECT ON SECRET2 TO CHRIS`.
Later Ann enters: `REVOKE SELECT ON SECRET2 FROM BILL`.
Let us assume that Oracle is used in this example. What is the result?

- ☐ Bill loses the `SELECT` right, Chris keeps it.
- ☐ Bill and Chris both lose the `SELECT` right.
- ☐ There is no way to revoke a right which was given with grant option.

c) Now Ann creates a table `SECRET3` and executes the following command:

```
GRANT SELECT ON SECRET3 TO BILL.
```

Bill executes the command

```
CREATE VIEW MY_SECRET AS SELECT * FROM SECRET3
```

and then `GRANT SELECT ON MY_SECRET TO CHRIS`.
Is it possible to share the secret in this way with Chris?

- ☐ No.
- ☐ Yes.

Results (To be Filled Out by the Instructor):

Exercise	Points	Max Points
1 (Translation ER to Relational Model)		10
2 (SQL CREATE TABLE)		5
3 (SQL)		18
4 (FDs, BCNF)		6
5 (Security)		3
Total		42