Dr. Stefan Brass
School of Information Sciences
University of Pittsburgh

April 24, 2000

INFSCI 2710 "Database Management" — Solution to Example Final Exam I —

Exercise 1 (SQL Queries) 16 Points

a)
```
SELECT P.ID, P.Path
FROM   Page P, Keyword X, Keyword Y
WHERE  X.ID = P.ID AND X.Word = 'Databases'
AND    Y.ID = P.ID AND Y.Word = 'WWW'
```

I like to write a tuple variable in front of every attribute. But this is a matter of personal taste. The tuple variable name is not required in front of "`Path`", because this attribute can only refer to "`P`".

b)
```
SELECT P.ID, P.Path
FROM   Page P
WHERE  P.ID NOT IN (SELECT R.ID
                    FROM   Request R
                    WHERE  R.when >= '01-JAN-99')
```

The exam was in 1999. Otherwise we would need to add `when <= '31-DEC-99'` in order to enforce "not requested in 1999".

c)
```
SELECT COUNT(DISTINCT Word)
FROM   KEYWORD
```

d)
```
SELECT   P.ID, P.Path, COUNT(*)
FROM     Page P, Request R
WHERE    P.ID = R.ID
GROUP BY P.ID, P.Path
HAVING   COUNT(*) >= 10
ORDER BY 3 DESC
```

An alternative is to define a name for the count: "`SELECT ..., COUNT(*) NUM_REQ`". Then you can say "`ORDER BY NUM_REQ DESC`".

Exercise 2 (Views) 6 Points

a)
```
CREATE VIEW UserHits AS
            SELECT   P.ID, P.Path, COUNT(*) HITS
            FROM     Page P, Request R
            WHERE    P.ID = R.ID AND P.Owner = USER
            GROUP BY P.ID, P.Path
            UNION
            SELECT   P.ID, P.Path, 0 HITS
            FROM     Page P
            WHERE    P.Owner = USER
            AND      P.ID NOT IN (SELECT R.ID FROM Request R)
```

It is possible to use "`UNION ALL`" here, which would actually make the query run a bit faster.

An alternative with an outer join is (in Oracle notation):

```
CREATE VIEW UserHits(ID, Path, Hits) AS
            SELECT P.ID, P.Path, COUNT(R.No)
            FROM   Page P, Request R
            WHERE  P.ID = R.ID(+)
            AND    P.Owner = USER
```

b) The view is not updatable because it contains an aggregation (`COUNT`). In most systems, the `UNION` and the join are also reasons for not allowing updates.

Exercise 3 (ER to Relational Mapping) 10 Points

```
CREATE TABLE Cab(
             No  NUMERIC(3) NOT NULL
                 CONSTRAINT Cab_No_Unique PRIMARY KEY,
             Bus CHAR(1) NOT NULL
                 CONSTRAINT Bus_Y_or_N_1 CHECK(Bus IN('Y','N')));
CREATE TABLE Goal(
             Phone   NUMERIC(10) NOT NULL
                     CONSTRAINT Goal_Key PRIMARY KEY,
             Name    VARCHAR(20) NOT NULL,
             Address VARCHAR(40) NOT NULL,
             ZIP     NUMERIC(5) NOT NULL);
```

```
CREATE TABLE Request(
            ID          NUMERIC(8) NOT NULL
                        CONSTRAINT Request_Key PRIMARY KEY,
            received    DATE DEFAULT SYSDATE NOT NULL,
            needed_at   DATE NOT NULL,
            Bus         CHAR(1) NOT NULL
                        CONSTRAINT Bus_Y_or_N_2 CHECK(Bus IN('Y','N')),
            Phone       NUMERIC(10) NOT NULL
                        CONSTRAINT Request_Ref_Goal REFERENCES Goal,
            No          NUMERIC(3) NULL
                        CONSTRAINT Valid_Cab_No REFERENCES Cab,
            CONSTRAINT Received_Before_Needed
                        CHECK(received < needed_at));
```

Exercise 4 (Integrity Constraints, Trigger) 5 Points

a)
```
SELECT R.ID
FROM   Request R, Cab C
WHERE  R.No = C.No
AND    R.Bus = 'Y' AND C.Bus = 'N'
```

b) The critical operations for this constraint are (only the first two had to be listed):

- `INSERT INTO Request`
  The constraint can be violated by an insertion into `Request` if the cab is immediately assigned: It is possible that the customer requested a bus, but the assigned cab is not a bus.
- `UPDATE Request SET No`
  When assigning or changing a cab for a request it is possible that a cab of the wrong type is chosen.
- `UPDATE Request SET Bus`
  This is already unlikely and was not required: It is conceivable that the customer calls again and changes his/her request into a request for a bus. This would also lead to a violation, if previously a normal taxi was assigned and this is not changed at the same time.
- `UPDATE Cab SET Bus`
  It is probably not possible that a bus suddenly becomes a normal taxi (unless the same cab number is used by different cars.) So it is not necessary to consider this update.