# Midterm Exam

Instructor: Taewhi Lee
April 22, 2013

1. [10pts] Describe two fundamental differences between the relational algebra and SQL.
Relational algebra: set-based (no duplicate tuple), procedural language
SQL: multiset-based (may have duplicate tuples), non-procedural language

2. [5pts] Explain why the following query does not work.

```
select   dept_name, id, count(name)
from     instructor
group by dept_name;
```

*id* cannot be used in the **select** clause
because *id* is not included in the **group by** expression.

3. [10pts] List five integrity constraints and briefly explain each of them.
**not null**: a specified attribute cannot be null
**unique**: values for specified attributes must be unique or null
**primary key**: values for specified attributes must be unique and not null
**check**: tuples must meet a given predicate
**foreign key**: a value that appears in one relation for a given set of attributes also appears for a certain set of attributes in another relation

4. a) [5pts] What is the difference between a relation and a view?
A view definition causes the saving of an expression;
the expression is substituted into queries using the view.
b) [5pts] Describe ~~three~~ advantages of using a view that joins two relations, against creating another relation that joins two relations.
~~Certain data can be hidden from the view of certain users.~~
No additional space overhead
Data modifications(insert/delete/update) are needed only for the underlying relations.

5. a) [5pts] Explain two major pitfalls to avoid in designing a database schema.
Redundancy: repeating information may cause data inconsistency
Incompleteness: it is difficult or impossible to model certain aspects of the enterprise
b) [5pts] Describe pros and cons of database normalization.
Pros: we can reduce data redundancy without loss of information.
Cons: more join operations between multiple tables are needed for data retrieval.
c) [5pts] Why do we sometimes need to denormalize a database schema instead of avoiding the pitfalls?
We may want to use non-normalized schema for performance, for example, to reduce join operations between multiple relations.

6. [5pts: 1pt each / 0pt for no answer / -1pt for a wrong answer]
Consider the relation r below. Which of the following FDs does r satisfy?
Choose TRUE or FALSE for each FD.

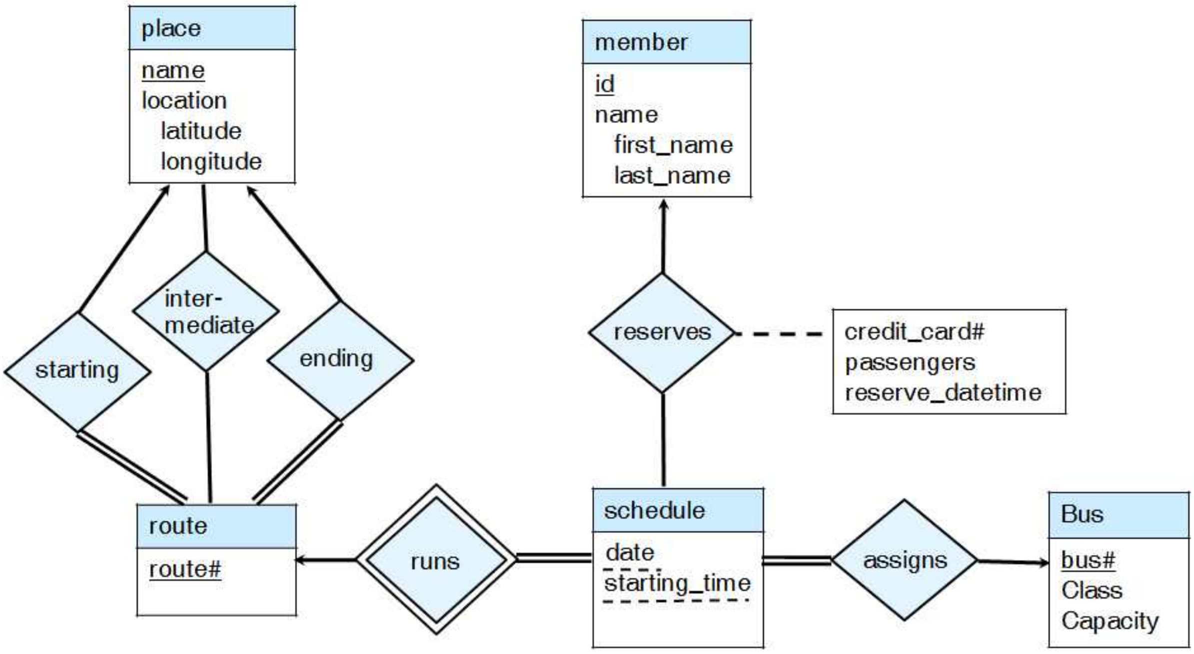
a) $A \rightarrow D$ FALSE
b) $AB \rightarrow D$ TRUE
c) $C \rightarrow BDE$ TRUE
d) $E \rightarrow A$ FALSE
e) $A \rightarrow E$ TRUE

Relation r

| A | B | C | D | E |
|---|---|---|---|---|
| $a_1$ | $b_1$ | $c_1$ | $d_1$ | $e_1$ |
| $a_1$ | $b_2$ | $c_2$ | $d_2$ | $e_1$ |
| $a_2$ | $b_1$ | $c_3$ | $d_3$ | $e_1$ |
| $a_2$ | $b_1$ | $c_4$ | $d_3$ | $e_1$ |
| $a_3$ | $b_2$ | $c_5$ | $d_1$ | $e_1$ |

7. a) [10pts] Construct an E-R diagram for the following description.

Design a database for the reservation office of a bus company.
- Each bus has a unique number. We also store its class and capacity.
- Each place has a unique name and location information of the latitude and longitude.
- Routes have a starting place and an ending place; also, some of them have several intermediate places.
- A number of buses are scheduled to a route. A bus is assigned to one schedule; some buses can have multiple schedule. We store the date and starting time of each schedule.
- A member of our company can book a bus by specifying a schedule. We store a unique id, first name, and last name of each member.
- For each reservation, credit card number, the number of passengers, and the reservation datetime are stored.




b) [10pts] Convert your E-R diagram into a relational schema.

place(<u>name</u>, latitude, longitude)
route(<u>route#</u>)
starting_place(<u>route#</u>, name)
ending_place(<u>route#</u>, name)
intermediate_place(<u>route#, name</u>)
schedule(<u>route#, day, starting_time</u>)
bus(<u>bus#</u>, class,capacity)
assignment(<u>route#, day, starting_time</u>, bus#)
member(<u>id</u>, first_name, last_name)
reservation(<u>id, route#, day, starting_time, credit_card#, passengers, reserve_datetime</u>)

8. Let F = { AB → C, B → D, CD → E, CE → GH }.
Give a derivation sequence on F for the following FDs using only Armstrong's axioms.
a) [5pts] AB → E
AB → ABD (augmentation: B → D with AB)
ABD → CD (augmentation: AB → C with D)
AB → CD (transitivity: AB → ABD and ABD → CD)
AB → E (transitivity: AB → CD and CD → E)
b) [5pts] AB → G
AB → ABE (augmentation: AB → E with AB)
ABE → CE (augmentation: AB → C with E)
AB → CE (transitivity: AB → ABE and ABE → CE)
AB → GH (transitivity: AB → CE and CE → GH)
GH → G (reflexivity)
AB → G (transitivity: AB → GH and GH → G)

9. Let F be the set of functional dependencies on relation schema R (A, B, C, D, E, F).
F = { A → B, C → D, AC → E, D → F}
a) [5pts] Find all candidate keys for the relation schema R.
Since AC → ABCDEF, AC is a superkey.
A and C must be in any candidate key since they do not appear on the right of any FD.
Hence, AC is the only candidate key.
b) [5pts] Give a lossless-join decomposition of R into BCNF.
R1(A, B), R2(C, D), R3(A, C, E), R4(C, F)
or R1(A, B), R2(C, D), R3(A, C, E), R4(D, F)
c) [5pts] Give a decomposition of R into 3NF, having a lossless-join and preserving dependencies. Is your decomposition is in BCNF?
R1(A, B), R2(C, D), R3(A, C, E), R4(D, F)
The answer is in BCNF.