

Midterm Exam

Instructor: Taewhi Lee
April 22, 2013

- [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
- [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.
- [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
- [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.
 - [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.
- [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
 - [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.
 - [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.
- [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 |

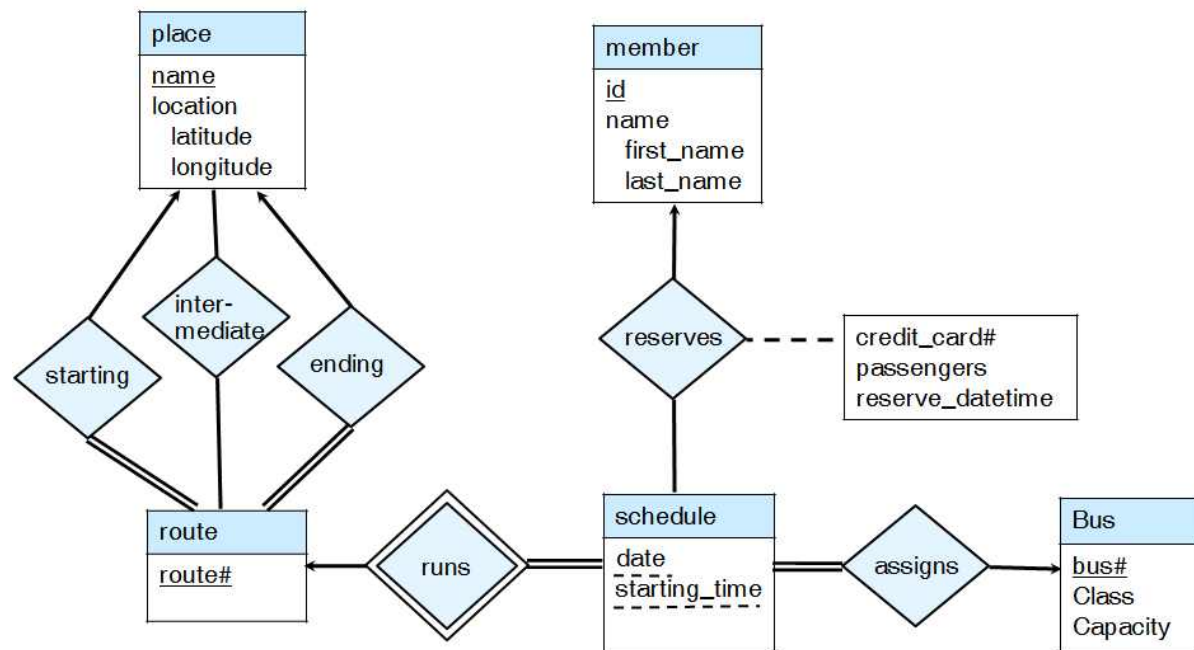
Relation *r*

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(name, latitude, longitude)

route(route#)

starting_place(route#, name)

ending_place(route#, name)

intermediate_place(route#, name)

schedule(route#, day, starting_time)

bus(bus#, class, capacity)

assignment(route#, day, starting_time, bus#)

member(id, first_name, last_name)

reservation(id, route#, day, starting_time, credit_card#, passengers, reserve_datetime)

8. Let $F = \{ AB \rightarrow C, B \rightarrow D, CD \rightarrow E, CE \rightarrow GH \}$.

Give a derivation sequence on F for the following FDs using only Armstrong's axioms.

a) [5pts] $AB \rightarrow E$

$AB \rightarrow ABD$ (augmentation: $B \rightarrow D$ with AB)

$ABD \rightarrow CD$ (augmentation: $AB \rightarrow C$ with D)

$AB \rightarrow CD$ (transitivity: $AB \rightarrow ABD$ and $ABD \rightarrow CD$)

$AB \rightarrow E$ (transitivity: $AB \rightarrow CD$ and $CD \rightarrow E$)

b) [5pts] $AB \rightarrow G$

$AB \rightarrow ABE$ (augmentation: $AB \rightarrow E$ with AB)

$ABE \rightarrow CE$ (augmentation: $AB \rightarrow C$ with E)

$AB \rightarrow CE$ (transitivity: $AB \rightarrow ABE$ and $ABE \rightarrow CE$)

$AB \rightarrow GH$ (transitivity: $AB \rightarrow CE$ and $CE \rightarrow GH$)

$GH \rightarrow G$ (reflexivity)

$AB \rightarrow G$ (transitivity: $AB \rightarrow GH$ and $GH \rightarrow G$)

9. Let F be the set of functional dependencies on relation schema $R(A, B, C, D, E, F)$.

$F = \{ A \rightarrow B, C \rightarrow D, AC \rightarrow E, D \rightarrow F \}$

a) [5pts] Find all candidate keys for the relation schema R .

Since $AC \rightarrow ABCDEF$, 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.

$R_1(A, B), R_2(C, D), R_3(A, C, E), R_4(C, F)$

or $R_1(A, B), R_2(C, D), R_3(A, C, E), R_4(D, F)$

c) [5pts] Give a decomposition of R into 3NF, having a lossless-join and preserving dependencies.

Is your decomposition in BCNF?

$R_1(A, B), R_2(C, D), R_3(A, C, E), R_4(D, F)$

The answer is in BCNF.