CS411
Database Systems

08: Midterm Review

Kazuhiro Minami

Midterm Details

- 3:30pm next Tuesday on October 13
- 75 minutes (regular class time slot)

- **Please arrive early**
- **Bring UIUC photoid, pen, pencil, eraser**
- **Closed book, no notes**

Questions during exam

- We will not answer technical questions.
- You can ask non-technical clarification questions.

- If you are not clear about assumptions in a problem, state your own assumptions and answer accordingly.

Answers

- Put your NetID on every page
- Write your answers in the exam sheet
- Make sure your work is concise and clear
- Show your working

- Will not penalize for minor SQL syntax errors (this is subjective but in your favor)
- Your writing must be legible

Coverage

- ER Data Model
- Relational Data Model
- Relational Algebra
- SQL
- Constraints and Trigger

Suggested Method of Study

- Go over the lecture slides
- Read the textbook
- Work on problems in hw/lectures
- Work on sample exams (see course Web)

E/R Mode Basics

- What are three basic elements in E/R model? How do we represent them in E/R diagrams?
- How do we express a key of an entity set?
- What types of multiplicity constraints on relationships? How do we express each of them?
- What type of multiplicity constraint can we specify on a multi-way relationship?
- What is a weak entity set? How do we express a weak entity set and supporting relationships?
- What is a isa relationship? What's the difference from ordinary relationships?
- Where can you find the keys of subclasses in a isa hierarchy?
- In what situations do you want to define a set of attributes rather than introducing a new entity set?

E/R Diagrams to Relation Schemas

- How to translate an entity set into a relation?
- How to translate a relationship into a relation?
- How to translate a weak entity set?
- How to translate entity sets in a isa hierarchy?
  - Do you know three different strategies?
  - Do you know the advantages and disadvantages among them?
  - Do we need to translate isa relationships into relations?
- In what situations can we combine relations?

Relational Schema Design

- What is a keys and a super key?
- What is a functional dependency?
- How to compute the closure of a set of attributes given FDs?
- How to use
- How to determine keys given functional dependencies?
- How to use Armstrong's Axioms and compute a closure of FDs?
- What are update and deletion anomalies? Why are they bad?
- What is BCNF?
- How to decompose relation into BCNF?
- Is BCNF decomposition lossless?
- What is 3NF? How is it different from BCNF?
- Is 3NF decomposition lossless?
- What are tradeoffs between BCNF and 3NF?
- What's a multi-valued dependency? Is it related to a FD in some way?
- What is the relationship between BCNF and 4NF?

Boyce-Codd Normal Form

A relation R is in BCNF if whenever there is a nontrivial FD $A_1 \ldots A_n \rightarrow B$ for R, $\{A_1 \ldots A_n\}$ is a superkey for R.

An FD is *trivial* if all the attributes on its right-hand side are also on its left-hand side.

Functional Dependency is a function whose exact formula we don't know

- Suppose that we want to define a function, but don't know the exact formula (i.e., Y = f(X)).
- The best we can do is to store input-output pairs of the function in a table.

X	Y
1	2
2	2
3	4

- Since f is a function, the same value of X is always mapped to the same value of Y.
- Therefore, we don't want to have two tuples with the same value of X.

We have redundant data when the table stores information on two different functions

- Now let's try to store information on another function Z = g(Y) in the same table.

X	Y	Z
1	2	3
2	2	3
3	4	5

- A key constraint ensures that X is unique in each row.
- But, we don't have the same guarantee for Y.
- Notice that we have two duplicate tuples (Y, Z) = (2,3).
- This is a kind of redundancy we want to remove using BCNF decomposition.
- Notice that if function f is one-one, we do not have this problem.

Another Situation Having Redundant Data

- Here is a situation where we have redundancy even if a function f is one-one.
- Suppose that we maintain information on two functions Z = f(X,Y) and W = g(Y)

X	Y	Z	W
1	2	3	5
2	2	3	5
3	4	5	3

• Notice that we have two duplicate tuples (Y, W) = (2, 5).

If the parameters of a function *h* contains a key, the table stores *h* with no redundancy

- We consider the table maintaining information Y = f(X) and Z = g(Y) again.
- We consider that the table maintain information on another function Z = h(X,Y) such that h(X, Y) = g(Y).

X	Y	Z
1	2	3
2	2	3
3	4	5

•We don't see any redundancy for maintaining information on function h. Why?

• X has a different value in each row, and thus (X, Y) is unique in each row as well.

• This corresponds to the fact that a FD whose left side attributes are a super key is OK.

Decompose R into a set of relations in BCNF

- R(A, B, C, D) with FD's AB ➔ C, C ➔ D, and D ➔ A

First, check whether R is in BCNF.

A relation R is in BCNF if whenever there is a nontrivial FD $A_1 \ldots A_n \rightarrow B$ for R, $\{A_1 \ldots A_n\}$ is a superkey for R.

1. Find keys of R

- R(A, B, C, D) with FD's AB ➔ C, C ➔ D, and D ➔ A

Strategy: Compute the closure of every subset of attributes in R

$\{A\}^+ = \{A\}$
$\{B\}^+ = \{B\}$
$\{C\}^+ = \{C, D, A\}$
$\{D\}^+ = \{D, A\}$
$\{A, B\}^+ = \{A, B, C, D\}$
$\{A, C\}^+ = \{A, C, D\}$
$\{A, D\}^+ = \{A, D\}$
$\{B, C\}^+ = \{B, C, D, A\}$
$\{B, D\}^+ = \{B, D, A\}$
$\{C, D\}^+ = \{C, D, A\}$
$\{A, B, C\}^+ = \{A, B, C, D\}$
$\{A, B, D\}^+ = \{A, B, C, D\}$
$\{A, C, D\}^+ = \{A, C, D\}$
$\{B, C, D\}^+ = \{A, B, C, D\}$
$\{A, B, C, D\}^+ = \{A, B, C, D\}$

2. Check whether R is in BCNF

- R(A, B, C, D) with FD's AB ➔ C, C ➔ D, and D ➔ A

Compare attributes on the left of each rule with two keys {A,B} and {B, C}

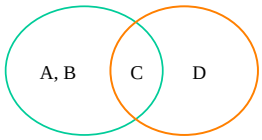
Q: Are there any FDs violating BCNF condition?

C ➔ D and D ➔ A

3. Decompose R

Let's pick C ➔ D and decompose R(A, B, C, D) into R1 and R2.

Q: What are the schemas of R1 and R2?



R1(A, B, C) R2(C, D)

Q: Is R2 in BCNF? Yes, but why?

Q: Is R1 in BCNF? We need to repeat the same process.

4. Find all non-tri… R1(A…

Do we only need to consider this FD?

- R1(A, B, C) with FD's AB ➔ C, C ➔ D, and D …

Compute th… …utes in R1

1. Start with {C}
2. Apply C ➔ D, and get {C, D}
3. Apply D ➔ A and get {C, D, A}
4. Project {C, D, A} onto {A, B, C} and get {C, A}

$\{A\}^+ = \{\sout{A}\}$
$\{B\}^+ = \{\sout{B}\}$
$\{C\}^+ = \{\sout{C,}\ A\}$
$\{A, B\}^+ = \{\sout{A, B,}\ C\}$
$\{A, C\}^+ = \{\sout{A, C}\}$
$\{B, C\}^+ = \{\sout{B, C,}\ A\}$
$\{A, B, C\}^+ = \{\sout{A, B, C}\}$

Q: What are non FD's?

$C \rightarrow A$,
$AB \rightarrow C$,
$BC \rightarrow A$

5. Decompose R1

- R1(A, B, C) with FD's $C \rightarrow A$, $AB \rightarrow C$, and $BC \rightarrow A$

Q: What are keys of R1? {A, B} and {B, C}.

Q: Which FD violate BCNF condition? $C \rightarrow A$

We decompose R1 into R11(A, C) and R12(B, C), which are in BCNF. Thus, R is decomposed into R2(C, D), R11(A, C), and R12(B, C).

Q: By the way, is R1 in 3NF? Yes because A is a prime.

When BCNF Decomposition Breaks FDs?

- We say that a decomposition is NOT dependency-preserving if we cannot check each FD with decomposed relations.

•Suppose that R(A, B, C) with FDs: A, B -> C, C->B.
•By using C -> B, we get R1(B, C) and R2(A,B).
•Therefore, we cannot check A,B -> C with R1 and R2.

3rd Normal Form

R is in 3NF
if for every nontrivial FD $A_1, ..., A_n \rightarrow B$,
either $\{A_1, ..., A_n\}$ is a superkey,
or B is part of a key.

Weakens BCNF.

Primary Goal of 3NF

- Preserve FDs of the initial relation R with docomposed relations $R_1,\ldots,R_n$.
- Try to minimize redundancy as long as the first goal is achieved.

Basic Approach for 3NF Decomposition

- Create a relation for each FD
  - E.g., If A->B, then create R(A, B)
- But, we want to minimize the number of such relations
- Thus, we find a minimum set of FDs, from which we can derive all FDs.
  - E.g., If A->B, B->C, we don't need to worry about A -> C. Thus, A->C is not part of the minimum set.
- All the FD in the minimum set has nice form:
  - $A_1,\ldots,A_n$ -> B where no FDs on the left and a single attribute on the right

Is a decomposed relation in BCNF?

- If $A_1,\ldots,A_n$ -> B in the minimum set, then we create $R(A_1,\ldots, A_n, B)$.
- There is no FDs among $A_1,\ldots,A_n$.
- What type of FD will violate BCNF?
- B -> $A_i,\ldots,A_j$ where $\{A_i,\ldots,A_j\} \subset \{A_1,\ldots,A_n\}$
- If we apply BCNF decomposition with the above FD, we break, $A_1,\ldots,A_n$ -> B will be broken.
- Recall the primary goal of 3NF.
- Then, let's stop here and find what property FD: B -> $A_i,\ldots,A_j$ has

Relational Algebra

- What are basic five operators in RA?
- What are derived operators?
- Do you know the symbols of the operators and what they do?
- How to define each derived operator with the basic ones?
  - What are theta-join and natural join?
- How to express the minimum value in R(a)?

Which products are available only at a single store?

R(ProductName, Store, ID)

ProductName	StoreID
Bread	2
Cheese-Cheddar	4
Cheese-Cheddar	5

Find all the store names whose products in their inventories are a subset of the inventory of some other store?

T(ProductName, StoreName)

ProductName	StoreName
Bread	WalMart
Cheese-Cheddar	Meijer
Cheese-Cheddar	Schnucks
Lettuce	Meijer

Your answer should be "Schnucks"

SQL

- What are the three clauses in a SQL query statement?
- How to express a single relation query in SQL with RA?
- Do you know how conditions involving NULL are evaluated?
- Do you know how to disambiguate attribute names in the WHERE clause when the FROM clause contain multiple relations?
- Do you understand the semantics (meaning) of a multi-relation query in SQL in two different ways?
- What if a query needs two copies of the same relation?
- In which clauses can we use subqueries?
- Can you use IN, ALL, ANY, and EXISTS operators on the result relation of a subquery?
- How to express set operations such as union, intersect, and set difference in SQL?
- Which operations in SQL support bag semantics?
- Which operations in SQL support set semantics?
- How to remove duplicate tuples from the result of a SQL query?
- Do you known when a correlated subquery is evaluated?

Online bookstore: Building a Recommendation Engine

Buy(tid, cid, isbn, year, month, day)

Q: Create a view FriendsOfBob that contains a list of people (i.e., a list of cid attribute values in Customer) who share a common interest with Bob whose cid = 12345. We consider that two persons share a common interest if they purchased more than 20 same books before.

Online bookstore: Building a Recommendation Engine

Buy(tid, cid, isbn, year, month, day), FriendsOfBob(cid)

Q: Make a list of recommended books (i.e., a list of isbn attribute values in Book) for Bob using view FriendsOfBob. We recommend a book for Bob if his possible friend in FriendsOfBob bought that book before and Bob has not bought that book yet.

SQL Aggregation/Grouping

- What are five aggregation function in SQL?
- How aggregations in SQL handle NULL values? Is there any difference among the functions?
- How to partition the result relation in a SQL query into multiple groups?
- Where can you define conditions on each group?
- Which attributes can you refer to in the HAVING clause?
- Which attributes can you include in the SELECT clause?
- In which clauses can you use aggregate functions?
- How to write insertion, deletion, and update statements in SQL?
- How to create a new table in SQL?
- What is a view?
- Why some views are not updatable?

Constraints and Triggers

- What's the major difference between constraints and triggers?
- What are example constraints in SQL?
- How primary keys and a set of attributes declared as UNIQUE handle NULL values differently?
- In which situations a foreign-key constraint could be violated?
- What are three strategies to prevent dangling tuples?
- When an attribute-based or tuple-based CHECK evaluated?
- In what situations a DBMS cannot enforce the conditions in CHECKs?
- When are the condition in an ASSERTION checked?
- What are events in a TRIGGER statement?
- What is the difference between statement-level triggers and row-level ones?