Northwestern University

Introduction to Database Systems

Sample Questions: Midterm

Mas-ud Hussain

Fall 2018

Types of Problems

- **Most Questions:** Similar to Homework/Project
  - May be a "bit" easier (or, less verbose) keeping in mind time constraints

- Additional Short/MCQ type questions

Problem Type 1: Text -> ERD

- Given a text snippets of project requirements specifications, draw an ERD.

Text -> ERD: Example

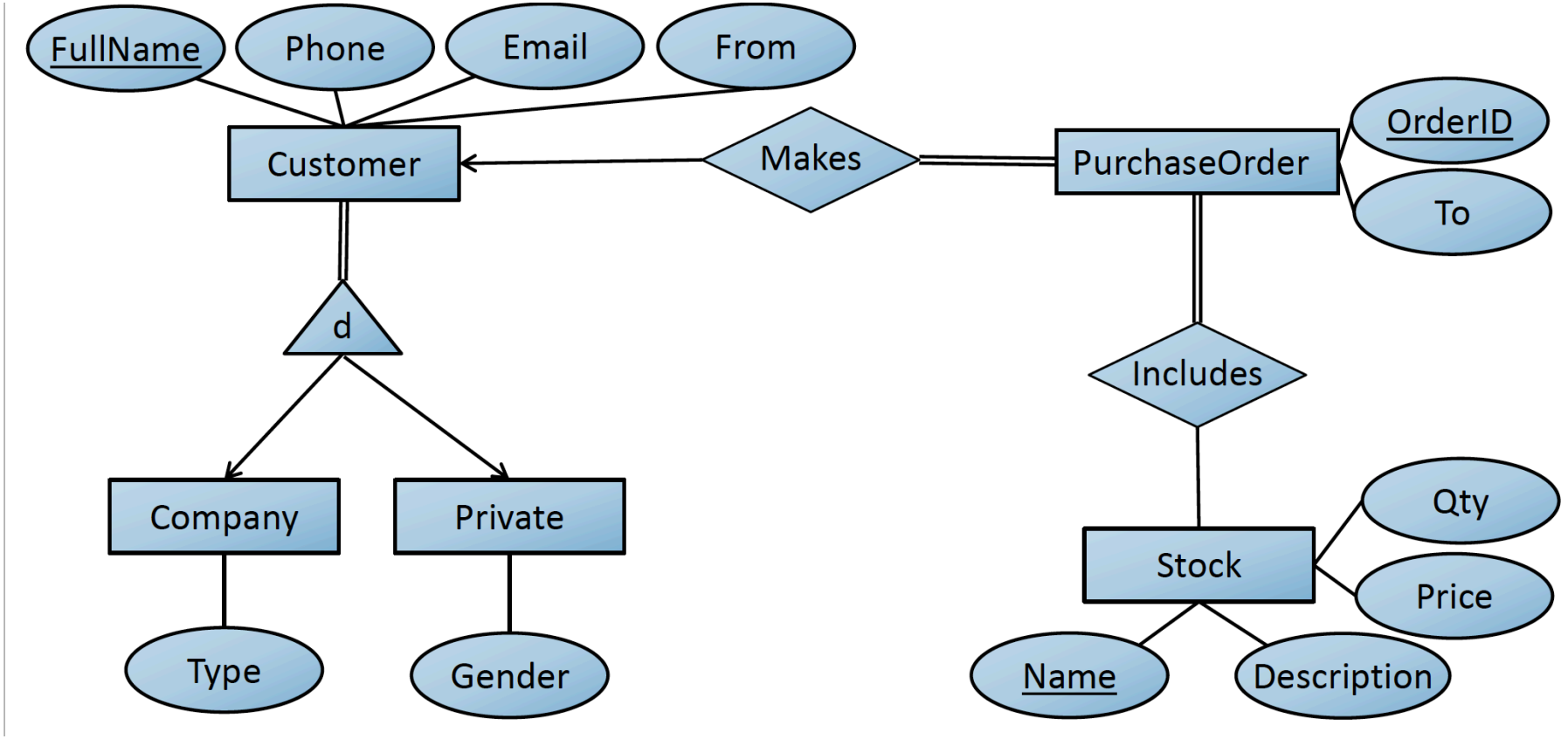
1 Flower shop

1(a). Imagine you are an owner of a flower shop. Design an ER diagram to represent your customers and purchase orders. Make sure to note the key attributes. The domain specifications are as follows:

- For each customer, you would store the full name of the customer, the "from" address, phone number and email address. No two customers have the same full name.
- The customer information is also divided into company customers and private customers. For each company customer, store the company name and business type. For each private customer, you store the customer's gender.
- For each purchase order, you would store the order ID, date, and "to" address.
- You also save information about your stock. Each stock item stores item quantity, price, name, and a description.

Additionally, any purchase order must be made by one customer only, and there must be at least one or more stocked items in a particular order.

Text -> ERD: Example



Text -> ERD: Notes

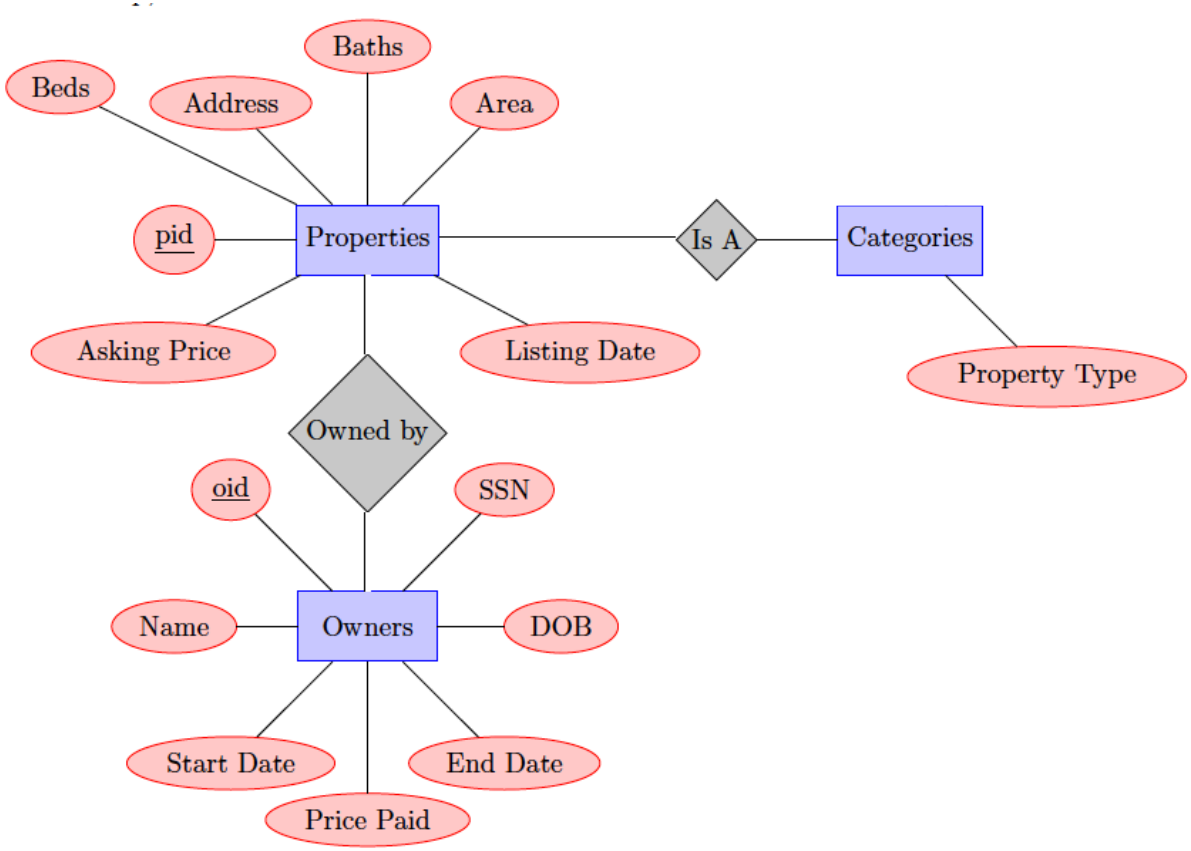
- Be careful about basic topics
  - Primary Keys
  - Relationship cardinality
  - Participation Constraints*
  - Weak entity or not?
  - ISA subclasses necessary or not? Disjoint or overlap?*
  - Ternary relationship or aggregation?
  - Multivalued/Composite attributes*

* whenever mentioned

Problem Type 2: ERD -> Relational Schema

- From a given ERD (or, part of it), produce the corresponding relational schema.
  - And/or, write SQL CREATE statements to materialize the given ERD.

ERD → Schema: Example



2. Using your entity-relationship diagram, create a schema for this database. Use the following format:

`tablename (attribute1, attribute2, attribute3)`

ERD → Schema: Example

- properties (pid, addr, area, beds, baths, ask_price, list_date, cid);
- category (pid, type);
- owners (oid, name, dob, ssn, start, end, price);
- owned_by (oid, pid);
  - Foreign Key: oid, pid

ERD -> Schema: Notes

- Follow the rules to convert specific ERD components to relational schema
  - 1-M, M-M, 1-1 relationships
  - ISA/Subclasses
  - Aggregation
  - Weak-entity
  - Specific Attributes
- Must point out foreign keys (if any), or any other necessary constraints (e.g., total participation == not null)

Problem Type 3: Short/MCQ questions

- Short questions about some particular topic.
  - Definitions
  - Testing basic knowledge

- MCQ questions

Short Questions: Example

Describe the three-tiered architecture of database-backed applications. Clearly state why we need each level.

- Web servers provide an interface with which users connect to the application. This manages the user interface and any resources (e.g., network ports) facing the outside world.
- Application servers execute the logic of an application. For example, a shopping cart application might keep track of a user's total dollar amount owed, update inventory for each purchase, and calculate the shipping costs associated with a transaction.
- Database servers efficiently query and store the data associated wtih an application.

Short Questions: Example

List two possible semantics that may be represented by a NULL value.

A NULL might imply that data is incomplete or missing. It might also mean that some property is not applicable to a given tuple (e.g., spouse field for a single person).

Short Questions: Example

Q1: SELECT R.A FROM R, S WHERE R.B = S.B;
Q2: SELECT R.A FROM R WHERE R.B IN (SELECT B FROM S);

(a) Q1 and Q2 produce the same answer.
(b) The answer to Q1 is always contained in the answer to Q2.
(c) The answer to Q2 is always contained in the answer to Q1.
(d) Q1 and Q2 produce different answers.

Short Questions: Example

For this problem assume that the schemas are R(B,A) and S(B,A) and B is a primary key in both relations.

Q1. $\pi_B(R-S)$

Q2: $\pi_B(R)-\pi_B(S)$

(e) Q1 and Q2 produce the same answer.
(f) The answer to Q1 is always contained in the answer to Q2.
(g) The answer to Q2 is always contained in the answer to Q1.
(h) Q1 and Q2 produce different answers.

Short Questions: Example

R

B	A
1	A
3	B
5	C
6	D

S

B	A
1	A
2	B
4	C
6	E

$\pi_B(R-S)$ **{3, 5, 6}**

$\pi_B(R)-\pi_B(S)$ **{3, 5}**

Short Questions: Example

Problem Define/explain *briefly* the following concepts:

Primary key

A primary key is a candidate key (minimal set of attributes that determine the values of all attributes in a relation) that is selected to represent the objects (rows) in the table. It is also a constraint on a relational schema that requires that the specified attributes be non-NULL and unique for every row.

Short Questions: Notes

- Don't be too verbose
  - 1-2 lines should be good enough
- MCQ: Look at questions carefully
  - If necessary, create sample scenarios and test your answer manually to be sure

- Short/MCQ questions can be on any topic!
- Sometimes a bunch of related questions can be grouped into one sequence of questions.

Problem Type 4: Write SQL Queries

- Given a particular relational schema, write appropriate SQL statements for the given queries.
  - A variant of these type of questions might also include a relational algebra expression (instead of query), and ask you to convert the expression to SQL

SQL: Example

7. Consider the following schema:

Suppliers(sid: integer, sname: string, location: string)
Parts(pid: integer, pname: string, color: string)
Catalog(sid: integer, pid: integer, price: real)

The Suppliers relation describes suppliers of parts. The Parts relation contains information about each part. The Catalog relation lists the prices in dollars charged for parts by suppliers. (The keys are underlined: sid is a key for Suppliers, (sid,pid) is a key for Catalog, and pid is a key for Parts.)

Write the following queries in SQL :

SQL: Example

7. Consider the following schema:

Suppliers(sid: integer, sname: string, location: string)
Parts(pid: integer, pname: string, color: string)
Catalog(sid: integer, pid: integer, price: real)

The Suppliers relation describes suppliers of parts. The Parts relation contains information about each part. The Catalog relation lists the prices in dollars charged for parts by suppliers. (The keys are underlined: sid is a key for Suppliers, (sid,pid) is a key for Catalog, and pid is a key for Parts.)

Write the following queries in SQL :

Find the names of parts that are supplied by exactly two suppliers.

```
SELECT pname
FROM Parts p, Catalog c, Suppliers s
WHERE p.pid = c.pid AND c.sid = s.sid
GROUP BY pid, pname
HAVING COUNT(*) = 2;
```

SQL: Example

7. Consider the following schema:

Suppliers(sid: integer, sname: string, location: string)
Parts(pid: integer, pname: string, color: string)
Catalog(sid: integer, pid: integer, price: real)

The Suppliers relation describes suppliers of parts. The Parts relation contains information about each part. The Catalog relation lists the prices in dollars charged for parts by suppliers. (The keys are underlined: sid is a key for Suppliers, (sid,pid) is a key for Catalog, and pid is a key for Parts.)

Write the following queries in SQL :

Find the names of the parts that are supplied by the most suppliers.

```
WITH suppCount AS (
        SELECT pname, COUNT(*) cnt
        FROM Parts p, Catalog c, Suppliers s
        WHERE p.pid = c.pid AND c.sid = s.sid
        GROUP BY pid, pname),
     maxSuppCount AS (
         SELECT max(cnt) mcnt
         FROM suppCount)
SELECT pname
FROM suppCount, maxSuppCount
WHERE cnt = mcnt;
```

SQL: Example

7. Consider the following schema:

Suppliers(sid: integer, sname: string, location: string)
Parts(pid: integer, pname: string, color: string)
Catalog(sid: integer, pid: integer, price: real)

The Suppliers relation describes suppliers of parts. The Parts relation contains information about each part. The Catalog relation lists the prices in dollars charged for parts by suppliers. (The keys are underlined: sid is a key for Suppliers, (sid,pid) is a key for Catalog, and pid is a key for Parts.)

Write the following queries in SQL :

Find the names and pids of parts that are supplied by every supplier at a price of strictly less than 200 dollars. (If a supplier does not supply the part or if a supplier charges more than 200 dollars for it, the part should not be output.)

```
WITH cheapParts AS (
       SELECT pid,pname, COUNT(*) cnt
       FROM Parts p, Catalog c, Suppliers s
       WHERE p.pid = c.pid AND c.sid = s.sid AND price < 200.0
       GROUP BY pid, pname),
     supplierCount AS (
       SELECT COUNT(*) allSupp FROM suppliers)
SELECT pid, pname
FROM cheapParts, supplierCount
WHERE cnt = allSupp;
```

SQL: Notes

- Do it as you have done in your project 1.
- Look at the wording carefully
  - "every", "most", "exactly 3", etc.
- If you are asked to convert a relational algebra expression to SQL
  - remember equivalent SQL keywords for each type of relational algebra operators

Problem Type 5: Relational Algebra Expressions

- Given a particular relational schema, write appropriate relational algebra expression for the given queries.

Relational Algebra: Example

Problem Assume that you are given a relational database capturing information about baseball games, teams, and players. The schemas of the corresponding relations are as follows:

```
Team(Name, City, Rank)

Game(TeamName1, TeamName2, Team1Score, Team2Score)

Player(Number, Name, TeamName)
```

Relational Algebra: Example

Problem Assume that you are given a relational database capturing information about baseball games, teams, and players. The schemas of the corresponding relations are as follows:

```
Team(Name, City, Rank)

Game(TeamName1, TeamName2, Team1Score, Team2Score)

Player(Number, Name, TeamName)
```

Write one or more *relational algebra expressions* that will retrieve the name of every team that has won at least one game.

$$\pi_{\text{Name}}(\sigma_{\text{Team1Score}>\text{Team2Score}}(Team \bowtie_{\text{Name} = \text{TeamName1}} Game) \cup \sigma_{\text{Team2Score}>\text{Team1Score}}(Team \bowtie_{\text{Name} = \text{TeamName2}} Game))$$

Relational Algebra: Example

Problem Assume that you are given a relational database capturing information about baseball games, teams, and players. The schemas of the corresponding relations are as follows:

```
Team(Name, City, Rank)

Game(TeamName1, TeamName2, Team1Score, Team2Score)

Player(Number, Name, TeamName)
```

Write an English description of the result of the relational algebra query

$$\pi_{\text{Name, Rank}}(Player \bowtie \rho_{\text{Name}\rightarrow\text{TeamName}}(Team))$$

Relational Algebra: Notes

- Do it as you have done in your homework 2.
- Look at the wording carefully
  - "every", "most", "exactly 3", etc.
- You can use the extra operators to perform aggregation/outer joins/group by for relational algebra if needed.

Problem Type 6: FD/Normal Forms

- Given a set of FDs F, check if another FD X->Y is in F+.

- Given a set of FDs F for a relation R, determine if R violates any normal form, e.g., 3NF, BCNF, etc.

FD: Example

Problem Consider the schema R(A, B, C, D, E, F, G, H).

Assume that the following functional dependencies have been identified:

1. $A \rightarrow DFG$
2. $BF \rightarrow E$
3. $CDG \rightarrow H$
4. $DH \rightarrow AE$
5. $FH \rightarrow BC$

For the next two parts of the problem, assume that the primary key has been selected as {A, H}. For each of the five functional dependencies, identify the strictest normal form (2NF, 3NF, or BCNF) to which it adheres. If a dependency violates 2NF, indicate that it only satisfies 1NF.

FD: Example

Problem Consider the schema R(A, B, C, D, E, F, G, H).

Assume that the following functional dependencies have been identified:

1. $A \rightarrow DFG$
2. $BF \rightarrow E$
3. $CDG \rightarrow H$
4. $DH \rightarrow AE$
5. $FH \rightarrow BC$

For the next two parts of the problem, assume that the primary key has been selected as {A, H}. For each of the five functional dependencies, identify the strictest normal form (2NF, 3NF, or BCNF) to which it adheres. If a dependency violates 2NF, indicate that it only satisfies 1NF.

$A \rightarrow DFG$ 1NF (violates 2NF)

2NF: Every attribute is fully functionally dependent on the primary key

FD: Example

Problem Consider the schema R(A, B, C, D, E, F, G, H).
Assume that the following functional dependencies have been identified:

1. $A \rightarrow DFG$
2. $BF \rightarrow E$
3. $CDG \rightarrow H$
4. $DH \rightarrow AE$
5. $FH \rightarrow BC$

For the next two parts of the problem, assume that the primary key has been selected as {A, H}. For each of the five functional dependencies, identify the strictest normal form (2NF, 3NF, or BCNF) to which it adheres. If a dependency violates 2NF, indicate that it only satisfies 1NF.

$BF \rightarrow E$ 2NF (violates 3NF)

R is in 3NF if, for all $X \rightarrow A$ in F^+,

- $A \in X$ (called a *trivial* FD), or
- X contains a key for R, or
- A is part of some key for R.

FD: Example

Problem Consider the schema R(A, B, C, D, E, F, G, H).
Assume that the following functional dependencies have been identified:

1. $A \rightarrow DFG$
2. $BF \rightarrow E$
3. $CDG \rightarrow H$
4. $DH \rightarrow AE$
5. $FH \rightarrow BC$

For the next two parts of the problem, assume that the primary key has been selected as {A, H}. For each of the five functional dependencies, identify the strictest normal form (2NF, 3NF, or BCNF) to which it adheres. If a dependency violates 2NF, indicate that it only satisfies 1NF.

$CDG \rightarrow H$ 3NF (violates BCNF)

Relation R with FDs *F* is in BCNF if, for all X → A in F^+

- A ∈ X (called a *trivial* FD), or
- X contains a key for R.

FD: Example

Let relation R(A,B,C,D,E) satisfy the following functional dependencies:

$AB \rightarrow C, BC \rightarrow D, CD \rightarrow E, DE \rightarrow A, AE \rightarrow B$

Show that, BC → A

FD: Example

Let relation R(A,B,C,D,E) satisfy the following functional dependencies:

$AB \rightarrow C, BC \rightarrow D, CD \rightarrow E, DE \rightarrow A, AE \rightarrow B$

Show that, BC → A.

(1) BC → D (Given)

(2) BC → CD (Augmentation, From 1)

(3) CD → E (Given)

(4) BC → E (Transitivity, From 2 & 3)

(5) BC → DE (Union, From 1 & 4)

(6) DE → A (Given)

(7) **BC → A (Transitivity, From 5 & 6)**

Relational Algebra: Notes

- Study the slides.

- Focus on:
  - Using the Armstrong Axioms, Union, Decomposition to derive desired FDs
  - The ideas of different normal forms
  - Simple decompositions

Problem Type 7: SQL -> Output

- Given a SQL query and the tables instances, write the output of the query.
  - In-place of SQL query, there can be relational algebra expressions as well

SQL->Output: Example

Consider the following tables:

Items

ItID	Desc	Price
101	Shag rug	89.97
102	Wool tapestry	154.99
103	Pewter bookend	26.00
104	Brass door knocker	59.98

Orders

CID	ItID	Qty
1	101	2
1	104	1
2	103	2
3	103	4

Customers

CID	Name	CCNum
1	Sharon Amity	90992
2	Joan Argent	29708
3	Paul Behr	36730

What is the output of the following SQL query:

```
SELECT Name, Qty * Price Total FROM Item
NATURAL JOIN Orders LEFT OUTER NATURAL JOIN Customers ORDER BY
Price, Qty;
```

SQL->Output: Example

Consider the following tables:

Items

ItID	Desc	Price
101	Shag rug	89.97
102	Wool tapestry	154.99
103	Pewter bookend	26.00
104	Brass door knocker	59.98

Customers

CID	Name	CCNum
1	Sharon Amity	90992
2	Joan Argent	29708
3	Paul Behr	36730

Orders

CID	ItID	Qty
1	101	2
1	104	1
2	103	2
3	103	4

What is the output of the following SQL query:

Name	Total
Joan Argent	52.00
Paul Behr	104.00
Sharon Amity	59.98
Sharon Amity	179.94

Rounding is fine, but the row order and column names should match the above.

```
SELECT Name, Qty * Price Total FROM Item
NATURAL JOIN Orders LEFT OUTER NATURAL JOIN Customers ORDER BY
Price, Qty;
```

SQL->Output: Notes

- Look carefully at the SQL/Relational Algebra queries.
  - You need to grasp the concept of different keywords and how they actually operate

- Is it a bag operation or set? Is DISTINCT used?
- Which join type is used? Natural? Outer?
- Affect of NULL values

Final Notes

- The exam is closed book/laptop/ipads.
  - Of course, open minds ☺
- You can bring (optionally) in a piece of normal A4 paper, where you can write things on one side of the page that you don't want to memorize
  - E.g., syntax, formal definition, keywords, etc.

Good Luck