# EECS 484

Discussion 6

Midterm Review

# Topics

- Functional Dependencies
- Relational Calculus
- SQL
- Relational Algebra

# Functional Dependencies

- Denoted by X-> Y
  - Y is functionally dependent on X
  - Values of Y are determined by X
- Primary key constraint is a special case of an FD
- Armstrong's axioms
  - Reflexivity: If $X \supseteq Y$, then $X \rightarrow Y$
  - Augmentation: If $X \rightarrow Y$, then $XZ \rightarrow YZ$ for any Z
  - Transitivity: If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$

# Normalization

- Normal forms
  - 3NF: if X -> A
    - $A \subseteq X$ (trivial dependency) or
    - X is a super key or
    - A is a part of some key
  - BCNF
    - $A \subseteq X$ (trivial FD), or
    - X is a super key
  - If an R is in BCNF, then it's in 3NF (not vice-versa)

- Decompositions: If R is decomposed into X and Y
  - Loss-less
    - $X \cap Y \rightarrow X$, or
    - $X \cap Y \rightarrow Y$
  - Dependency-preserving:
    - $F^+ = (F_x \cup F_y)^+$

# Functional Dependencies

Consider a relation R with attributes ABCD. Suppose the following are the only FDs for R.

$$A \longrightarrow B$$

$$BC \longrightarrow D$$

$$A \longrightarrow C$$

a) Identify all (minimal) *candidate keys* for R

Note: Use Armstrong's Axioms

# Functional Dependencies

Consider a relation R with attributes ABCD. Suppose the following are the only FDs for R.

$A \rightarrow B$

$BC \rightarrow D$

$A \rightarrow C$

a) Identify all (minimal) *candidate keys* for R

$A \rightarrow B$ and $A \rightarrow C$

**implies** $A \rightarrow BC$ (union)

$A \rightarrow BC$ and $BC \rightarrow D$

**implies** $A \rightarrow ABCD$ (transitivity + union)

# Functional Dependencies

Consider a relation R with attributes ABCD. Suppose the following are the only FDs for R.

$A \longrightarrow B$

$BC \longrightarrow D$

$A \longrightarrow C$

b) Are there any other candidate keys that do not include A?

# Functional Dependencies

Consider a relation R with attributes ABCD. Suppose the following are the only FDs for R.

$$A \longrightarrow B$$

$$BC \longrightarrow D$$

$$A \longrightarrow C$$

b) Are there any other candidate keys that do not include A?

No!

# Functional Dependencies

Consider a relation R with attributes ABCD. Suppose the following are the only FDs for R.

$A \longrightarrow B$

$BC \longrightarrow D$

$A \longrightarrow C$

c) Given the above FD, is R in Boyce-Codd Normal Form (BCNF)?

Note:

Relation R with FDs F is in BCNF if, for all $X \longrightarrow A$ in $F^+$

- $A \subseteq X$ (trivial FD), or
- X is a superkey

# Functional Dependencies

c) Given the above FD, is R in Boyce-Codd Normal Form (BCNF)?

- No!
- $BC \longrightarrow D$
  - This is not a trivial FD (D is not a subset of BC)
  - BC is not a superkey

# Functional Dependencies

d) Suppose that we decided to decompose R into two relations, R1 = BCD and R2 = ABC. Is this a lossless join decomposition?

Note:

Lossless-join w.r.t. F if and only if F+ contains:

- $R1 \cap R2 \rightarrow R1$, or
- $R1 \cap R2 \rightarrow R2$

That is, attributes common to R1 and R2 contain a key for either R1 or R2

# Functional Dependencies

d) Suppose that we decided to decompose R into two relations, R1 = BCD and R2 = ABC. Is this a lossless join decomposition?

- Yes!
- $R1 \cap R2 = BC$
- BC is a key for R1 since $BC \longrightarrow D$ implies $BC \longrightarrow BCD$

# Relational Calculus

Consider the following relational schema for a library (primary keys are underlined)

Member(<u>mid</u>, name, dob)

Book (<u>isbn</u>, title, author, publisher)

Borrow (<u>mid, isbn, date</u>)

# Relational Calculus

Consider the following relational schema for a library (primary keys are underlined)

Member(<u>mid</u>, name, dob)

Book (<u>isbn</u>, title, author, publisher)

Borrow (<u>mid, isbn, date</u>)

a) Write a relational calculus expression to find the names of members who have borrowed atleast one book published by McGraw-Hill

# Relational Calculus

a) Write a relational calculus expression to find the names of members who have borrowed atleast one book published by McGraw-Hill

$$\{T \mid \exists M \in Member, \exists B \in Book, \exists R \in Borrow\ ( \\ M.mid = R.mid \land B.isbn = R.isbn \\ \land B.publisher =' McGraw - Hill' \land T.name = M.name)$$

# Relational Calculus

Consider the following relational schema for a library (primary keys are underlined)

Member(<u>mid</u>, name, dob)

Book (<u>isbn</u>, title, author, publisher)

Borrow (<u>mid, isbn, date</u>)

b) Write a relational calculus expression to find the *mid* of members who have borrowed a book titled 'The Catcher in the Rye' and book titled 'A Farewell to Arms'

# Relational Calculus

b) Write a relational calculus expression to find the *mid* of members who have borrowed a book titled 'The Catcher in the Rye' and book titled 'A Farewell to Arms'

$$
\begin{aligned}
&\{T \mid \exists B1 \in Borrow, \exists B2 \in Borrow, \exists K1 \in Book, \exists K2 \\
&\in Book\ (B1.isbn = K1.isbn \\
&\wedge K1.title =' A\ Farewell\ to\ Arms' \\
&\wedge B2.isbn = K2.isbn \\
&\wedge K2.title =' The\ Catcher\ in\ the\ Rye' \\
&\wedge B1.mid = B2.mid \wedge T.mid = B1.mid)\}
\end{aligned}
$$

# SQL

Consider the following relational schema for drivers and cars (primary keys are underlined)

Driver (<u>licence_num</u>, name, age)

Car (<u>vin</u>, make, model, year)

Owns(<u>license_num</u>, <u>vin</u>)

# SQL

Consider the following relational schema for drivers and cars (primary keys are underlined)

Driver (<u>license_num</u>, name, age)

Car (<u>vin</u>, make, model, year)

Owns(<u>license_num</u>, <u>vin</u>)

a) Write a SQL query to find the names of all drivers under age 18 who own a car.

# SQL

a) Write a SQL query to find the names of all drivers under age 18 who own a car.

```
SELECT D.name
FROM Driver D, Owns O
WHERE D.license_num = O.license_num AND D.age < 18
```

# SQL

Consider the following relational schema for drivers and cars (primary keys are underlined)

Driver (license_num, name, age)

Car (vin, make, model, year)

Owns(license_num, vin)

b) Write a SQL query to find the makes of cars where the average owner is > 65. For example, if the average age of people who own a Cadillac is 72, then the result set should contain the value Cadillac.

# SQL

b) Write a SQL query to find the makes of cars where the average owner is > 65. For example, if the average age of people who own a Cadillac is 72, then the result set should contain the value Cadillac.

```
SELECT C.make
FROM Owns O, Car C, Driver D
WHERE O.vin = C.vin
AND D.license_num = O.license_num
GROUP BY C.make
HAVING AVG(D.age) > 65
```

# SQL

Consider the following relational schema for drivers and cars (primary keys are underlined)

Driver (<u>license_num</u>, name, age)

Car (<u>vin</u>, make, model, year)

Owns(<u>license_num</u>, <u>vin</u>)

c) Write a SQL query to find the vin of cars that are older (have an earlier year) than all cars with make "Ferrari"

# SQL

c) Write a SQL query to find the vin of cars that are older (have an earlier year) than all cars with make “Ferrari”.

```
SELECT C.vin
FROM Cars C
WHERE C.year < ALL (
    SELECT C2.year
    FROM Cars C2
    WHERE C2.make = ‘Ferrari’)
```

# Relational Algebra

You are given the following relations (primary keys are underlined)

Student (<u>sid</u>, name, age)

Take(<u>sid</u>, <u>cid</u>, grade)

Course (<u>cid</u>, title)

a) Write a relational algebra expression to find the *sid* of the oldest student. If there is a tie, you should return the *sid* of all students with maximum age

# Relational Algebra

- Write a relational algebra expression to find the sid of the oldest student. If there is a tie, you should return the sid of all students with maximum age

$$\pi_{sid}(Students)$$
$$-\ \pi_{S1.sid}(Students\ S1 \bowtie_{S1.age<S2.age} Students\ S2)$$

- NOTE: There is no “MAX” function in relational algebra

# Main exam topics

- ER modeling
- SQL
- Relational algebra
- Relational calculus
- Application programming
- Normalization using functional dependencies