

CS411 Database Systems

Fall 2004, Prof. Chang

Department of Computer Science
University of Illinois at Urbana-Champaign

Midterm Examination

October 22, 2004

Solution

Problem 1 (*15 points*) Misc. Concepts

- (1) True
- (2) False
- (3) True
- (4) True
- (5) True
- (6) True
- (7) True
- (8) False
- (9) True
- (10) False
- (11) False
- (12) True
- (13) False
- (14) False
- (15) True

Problem 2 (*20 points*) ER and Schema Design

- (a) (*6 points*)
 - FD1: *Model* \rightarrow *Make*
 - FD2: *Model, OwnerSSN* \rightarrow *YearBought, Make*
 - FD3: *DealerName* \rightarrow *Address, Zip*

(b) (6 points)

FD4: $OwnerSSN, Model \rightarrow DealerName$

(from FD2, FD3, and the many-to-one relationship from Car to Dealership)

FD5: $Model, OwnerSSN \rightarrow Address, Zip$

(from FD3 and FD4)

And many more "trivial" ones by applying Reflexivity, Augmentation, and Transitivity rules.

(c) (8 points)

For Car entity set: $(Model, Make), (OwnerSSN, Model, YearBought)$

For Dealership entity set: $(DealerName, Zip, Address)$

For the many-to-one relationship: $(OwnerSSN, Model, DealerName)$

Note: $(OwnerSSN, Model, YearBought)$ and $(OwnerSSN, Model, DealerName)$ can be merged.

Problem 3 (20 points) Query Languages

(a) (6 points)

$\pi_{barName}(\sigma_{drinkName='Sally'}(Frequent))$

(b) (7 points)

$\pi_{drinkName}(Frequent)$

– $\pi_{drinkName}(Frequent \bowtie (Drinker \bowtie_{Drinker.street \neq Bar.street} Bar))$

Alternative solution:

$\pi_{drinkName}(Frequent)$

– $\pi_{drinkName}(\sigma_{Drinker.street \neq Bar.street}(Drinker \bowtie Frequent \bowtie Bar))$

(c) (7 points)

SELECT *barName*

FROM *Drinker, Frequent*

WHERE *Drinker.drinkerName = Frequent.drinkerName*

GROUP BY *barName*

HAVING AVG(age) < 37

Problem 4 (20 points) Views and Constraints

(a) (4 points)

branchName of *Account* relation referencing *branchName* of *Branch* relation

customerSSN of *Deposit* relation referencing *customerSSN* of *Customer* relation

accountNumber of *Deposit* relation referencing *accountNumber* of *Account* relation

(b) (8 points)

```
CREATE VIEW BigBranch AS
SELECT Branch.branchName, city, assets
FROM Branch, Account
WHERE Branch.branchName = Account.branchName
GROUP BY Branch.branchName, city, assets
HAVING COUNT(accountNumber) > 50
      AND SUM(balance) > 1,000,000)
```

(c) (8 points)

```
NOT EXISTS
(SELECT Branch.branchName
FROM Account, Branch
WHERE Account.branchName = Branch.branchName
GROUP BY Branch.branchName
HAVING SUM(balance) > assets)
```