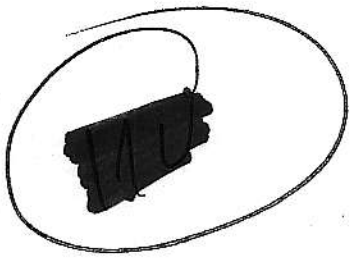
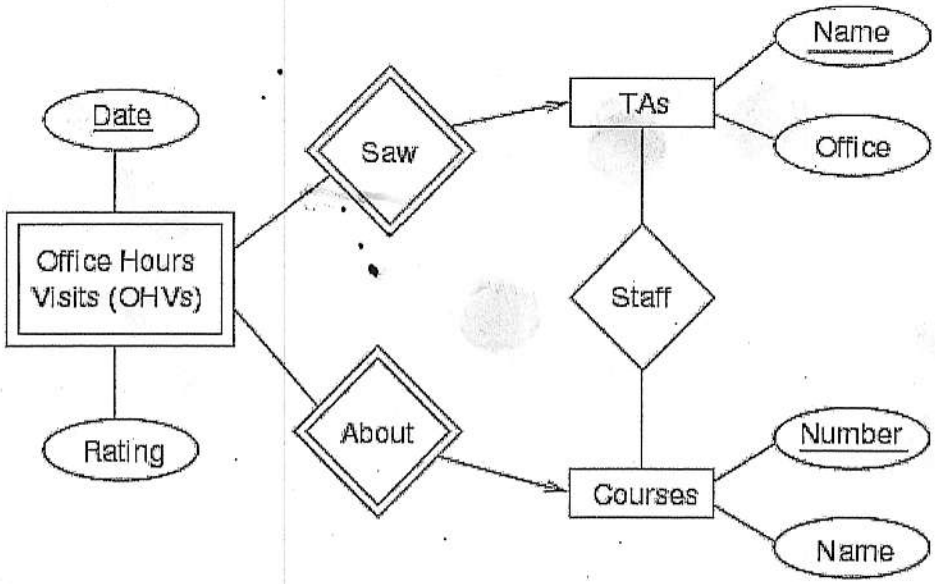
CSE 412/598
Database Management Systems
Mid-Term Exam
Spring 2007



NAME [redacted] **ASU ID:** [redacted]

Problem 1- **(25 points)** Consider the following ER diagram, which specifies courses, TAs for courses and office hours visits to TAs including a rating for the quality of each visit.



Perform a conversion of the E/R diagram into relation schemas. Be sure to specify (underline) keys for all relations. List your set of relations here:

(1) Entities

TAs (Name , office)
Courses (Number, Name)

(2) Weak Entities

OfficeHoursVisits (Name , Date , Rating) or OfficeHoursVisits (Number, Date, Rating)

(3) Relationships

Staff (Name , Number)

Problem 2 - **(30 points)** Consider the following relational schema:

Course(course#, dept_name)
Enroll(studentID, course#)

Answer these queries using the relational algebra:

1) **(7 points)** Find the studentID's of all students who are not enrolled in course# 145. (↗ difference)

$\Pi_{studentID}(Enroll) - \Pi_{studentID}(\sigma_{course\#=145}(Enroll))$

all students — students who enrolled in course# 145

2) **(8 points)** Find the studentID's of all students who are enrolled in courses in at least two different departments. (more than one.)

% Find courses in at least two different departments:
CourseCopy(course#2, dept_name2) := Course;
Course2Dept := $\Pi_{course\#}(\sigma_{course\#=course\#2 \text{ and } dept\_name <> dept\_name2}(Course \times CourseCopy))$;

% Find the required studentIDs
$q2 := \Pi_{studentID}(Course2Dept \bowtie_{course\#} Enroll)$;

3) **(7 points)** Find the studentID's of all students who are enrolled in courses from only one department (Hint. You can use your answer to (2) above.). (→ difference)

Method 1: % All students − Students enrolled in courses in at least 2 departments
$\Pi_{studentID}(Enroll) - q2$;

Method 2: % All courses − courses offered at ≥2 departments:
temp := $\Pi_{course\#}(Course) - Course2Dept$;
$q3 := \Pi_{studentID}(temp \bowtie Enroll)$;

4) **(8 points)** Find the studentID's who are enrolled in all courses. (Hint. You can use the division operator directly.)

% Enroll(studentID, course#) ÷ CourseID(course#) = StudentID

Enroll ÷ ($\Pi_{course\#}(Course)$)

Problem 3 – **(25 points)** Query Optimization
Consider the following relational schema and relational algebra query.

customer (cName, street, cCity)
deposit (cName, bName, accountNumber, balance)
brach (bName, assets, bCity)

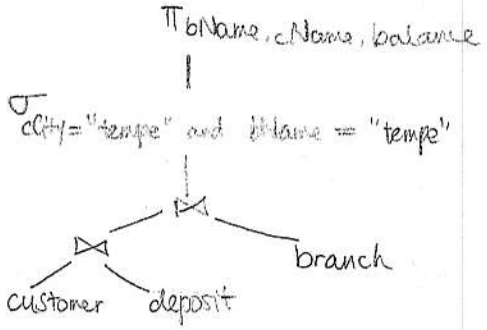
Identify a relational algebra tree that reflects the order of operations a decent query optimizer would choose. Use the following statistics to estimate the sizes of intermediate tables and total join costs.

Assume, |customer| = 1,000,
|deposit| = 1,000,
|branch| = 10,
V(cCity, customer) = 10,
V(bCity, branch) = 10,
Max(balance) = 100K

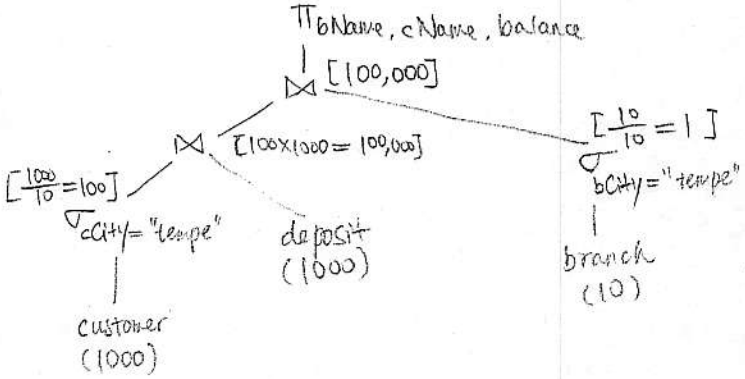
QUERY:

Π bName, cName, balance (σ cCity = "tempe" and ~~bName~~ bCity = "tempe" ((customer $\bowtie$ deposit) $\bowtie$ branch))

① Initial Query Tree

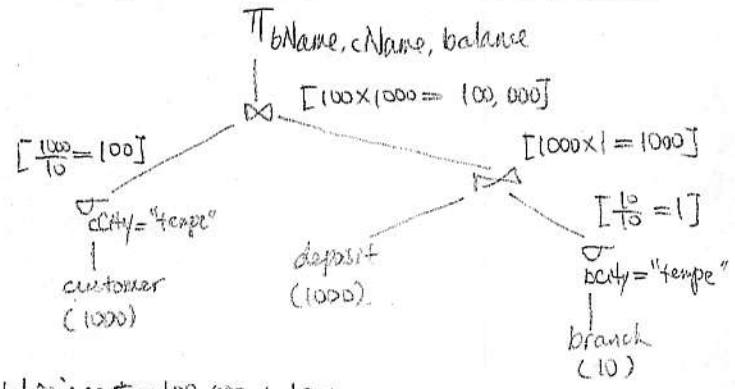


② Break down σ and do Estimation



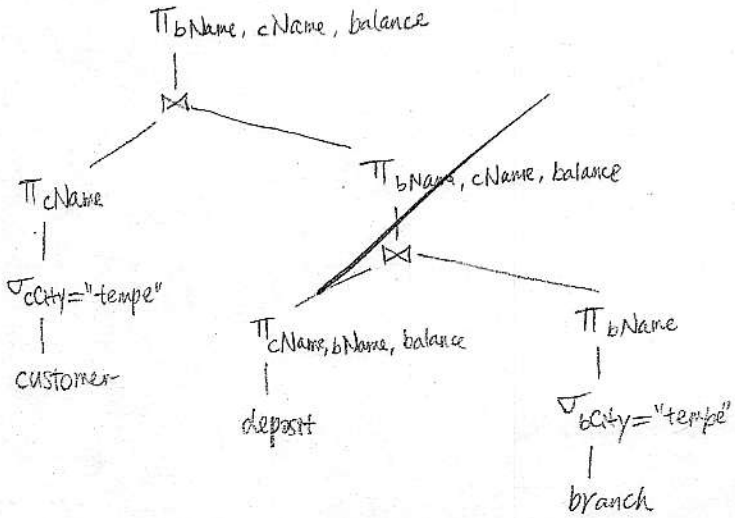
∴ total join cost = 100,000 + 100,000 = 200,000

③ Rearrange the tree and do Estimation



∴ total join cost = 100,000 + 1,000
= 101,000 < 200,000 ⇒ use this tree

④ Break down Π



Problem 4 – **(15 points)** Functional Dependencies for keys

Consider a relation R(A,B,C,D,E, F) with the following set of functional dependencies:
A → C, DE → F, B → D.

1) **(8 points)** What is the key for R?

$\{A,B,E\}^+ = \{A,B,E,C,D,F\} = R \Rightarrow \{A,B,E\}$ is the key for R

2) (7 **points**) Add to the above set of functional dependencies the dependency A → B. Now, suppose we want **A** to be a key. Name one more functional dependency with only one attribute on the left-hand side and only one attribute on the right-hand side, that if added to the set, makes **A** a key.

Given: $A \to B$, $B \to D$
$A \to C$
$DE \to F$

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

Hence, we can add any one of the following FDs to make $\{A\}^+ = R$.

① $A \to E$ ② $B \to E$ ③ $C \to E$, ④ $D \to E$

For example: $F = \{A \to B, B \to D, A \to C, DE \to F, ①\ A \to E\}$

$\{A\}^+ = \{A,B,C,E,D,F\} = R$ ∴ A is a key

Problem 5 – **(15 points)** Functional Dependencies

1) **(8 points)** Consider a relation schema R=(A,B,C,D,E) and a set F consisting of the following functional dependencies that hold on R: A → C, AC → E, AD → B, B → ADE, D → E. Is it true that F^+, the *closure* of F, contains the functional dependencies (a) A → E and (b) B → C?

(a) $\{A\}^+ = \{A,C,E\} \supset \{E\}$

Method 2:

result: A, (given)
C, (by A → C)
E, (by AC → E)

∴ Yes. A → E is in F^+

(b) result: B (given)
A, D, E (by B → A, D, E)
C, (by A → C)

∴ Yes.
B → C is in F^+

2) (7 **points**) Determine whether or not the decomposition of R into the relation schemas R_1=(A,C,D) and R_2=(A,B,E) is lossless-join under F. Justify your answer.

By given $R = \{A,B,C,D,E\}$, $F = \{A \to C, AC \to E, AD \to B, B \to ADE, D \to E\}$

	A	B	C	D	E
$R_1=(A,C,D)$	✓		✓	✓	②✓
$R_2=(A,B,E)$	✓	✓	①✓		✓

First loop
①✓ A → C
②✓ AC → E

Second loop
add nothing.

∵ there is no completed row
∴ This is NOT lossless-join,
that is, R_1 and R_2 is not lossless-join under