9. 관계 데이타 연산

관계 데이타 연산(시스템)

- 관계 데이타 언어(사용자)
  - i . 관계 대수(relational algebra)
    - 절차언어 : how, what
  - ii . 관계해석(relational calculus)
    - 비절차언어 : what
    - 투플 관계해석
    - 도메인 관계해석

- 관계 해석과 관계 대수는 표현이나 기능면에서 동등

- 상용 관계 데이터 언어
  - 관계 대수와 관계 해석을 기초로 함.
  - SQL, QUEL, QBE

9.1 관계대수(Relational Algebra)

- 릴레이션 조작을 위한 연산의 집합
- 폐쇄성질 (closure property)
  - 피연산자와 연산 결과가 모두 릴레이션
  - 중첩(nested)된 수식의 표현이 가능
- 구성
  - 릴레이션 : 투플의 집합
  - 일반 집합 연산자 : 합집합(union)
교집합(intersection)
차집합(difference)
카티션 프로덕트(cartesian product)
  - 순수 관계 연산자 : 실렉트(select)
프로젝트(project)
조인(join)
디비전(division)

9.1.1 일반 집합 연산자

- 합집합 (union, $\cup$)
  - 정의: $R \cup S = \{ t \mid t \in R \vee t \in S \}$
  - Cardinality: $\max\{|R|,|S|\} \le |R \cup S| \le |R| + |S|$
- 교집합 (intersect, $\cap$)
  - 정의: $R \cap S = \{ t \mid t \in R \wedge t \in S \}$
  - Cardinality: $0 \le |R \cap S| \le \min\{|R|,|S|\}$
- 차집합 (difference, $-$)
  - 정의: $R - S = \{ t \mid t \in R \wedge t \notin S \}$
  - Cardinality: $0 \le |R - S| \le |R|$
- 카티션 프로덕트 (cartesian product, $\times$)
  - 정의: $R \times S = \{ r \cdot s \mid r \in R \wedge s \in S \}$
    단, $\cdot$: 접속(concatenation)
  - Cardinality: $|R \times S| = |R| \times |S|$
  - 차수(degree): R의 차수 + S의 차수

Note

- 합병가능(union-compatible)한 릴레이션
  - ∪, ∩, − 연산의 피연산자들은
    - i. 차수가 같아야 함
    - ii. 대응 애트리뷰트 별로 도메인이 같아야 함

- ∪, ∩, × 연산은 결합적(associative)임
  $A \cup B \cup C = (A \cup B) \cup C = A \cup (B \cup C)$

- ∪, ∩, × 연산은 교환적(commutative)임
  $A \cup B = B \cup A$

Example: 대학(University) 관계 데이타베이스

학생
(STUDENT)

학번 (SNO)	이름 (SNANE)	학년 (YEAR)	학과 (DEPT)
100	나 연 묵	4	컴퓨터
200	이 찬 영	3	전기
300	정 기 태	1	컴퓨터
400	송 병 호	4	컴퓨터
500	박 종 화	2	산공

과목
(COURSE)

과목번호 (CNO)	과목이름 (CNANE)	학점 (CREDIT)	학과 (DEPT)	담당교수 (PRNAME)
C123	프로그래밍	3	컴퓨터	김성기
C312	자료 구조	3	컴퓨터	황수찬
C324	파일 처리	3	컴퓨터	이규철
C413	데이타 베이스	3	컴퓨터	이석호
C412	반 도 체	3	전자	홍봉희

등록
(ENROL)

학번* (SNO)	과목번호* (CNO)	성적 (GRADE)	중간성적 (MIDTERM)	기말성적 (FINAL)
100	C413	A	90	95
100	E412	A	95	95
200	C123	B	85	80
300	C312	A	90	95
300	C324	C	75	75
300	C413	A	95	90
400	C312	A	90	95
400	C324	A	95	90
400	C413	B	80	85
400	E412	C	65	75
500	C312	B	85	80

9.1.2 순수 관계 연산자

- Symbolic notations
  - 릴레이션, R
    - $R(X)$, $X=\{A_1, \ldots, A_n\}$ ⟶ $R(A_1, \ldots, A_n)$
  - R의 투플, r
    - $r \in R$, $r = \langle a_1, \ldots, a_n \rangle$
  - 투플 r의 애트리뷰트 A_i의 값
    - $r.A_i = r[A_i] = a_i$

셀렉트 (SELECT, σ)

- 정의
  - A, B가 릴레이션 R(X)의 애트리뷰트일 때 (A∈X, B∈X),

$$\sigma_{A\theta v}(R) = \{ r \mid r \in R \wedge r.A \theta v \}$$
$$\sigma_{A\theta B}(R) = \{ r \mid r \in R \wedge r.A \theta r.B \}$$

조건식(predicate)

단, θ(theta) = {<, >, ≤, ≥, =, ≠}
v : 상수

- 선택 조건을 만족하는 릴레이션의 수평적 부분집합 (horizontal subset)

- Example
  - $\sigma_{\text{학과='컴퓨터'}}$ (학생)
  - $\sigma_{\text{학번=300} \wedge \text{과목번호='C312'}}$(등록)
  - $\sigma_{\text{중간성적<기말성적}}$ (등록)

- 데이타 언어식 표현
  - WHERE 조건식

- 실렉트 연산은 교환적(commutative)임
  - $\sigma_{\text{조건2}}(\sigma_{\text{조건1}}(R)) = \sigma_{\text{조건1}}(\sigma_{\text{조건2}}(R)) = \sigma_{\text{조건1} \wedge \text{조건2}}(R)$

- 선택도(selectivity)
  - 조건식에 의해 선택된 투플의 비율

프로젝트 (PROJECT, Π)

- 정의
  - 릴레이션 R(X), $X=\{A_1,A_2, \ldots ,A_n\}$에서
    애트리뷰트 집합 Y가 $Y \subseteq X$, $Y=\{B_1,B_2, \ldots ,B_m\}$, $n \geq m$ 이면,

    $\Pi_Y(R) = \{ \langle r.B_1, \ldots , r.B_m \rangle \mid r \in R \}$

- example
  - 학생(학번,이름,학년)에서 $\Pi_{이름}$(학생)

- 릴레이션의 수직적 부분집합(vertical subset)

- 생성된 중복 투플은 제거

- $\Pi_Y(\Pi_X(R)) = \Pi_Y(R)$

예제 : 셀렉트와 프로젝트

학번	이름	…	…	학과	…	학년
1	가					
2	나					
3	다					
4	라			CS		3
5	마			CS		3
6	라			CS		3
7	바					
8	사					
9	나					

셀렉트

프로젝트

학번	이름	…	…	학과	…	학년
4	라			CS		3
5	마			CS		3
6	라			CS		3

학번	이름
1	가
2	나
3	다
4	라
5	마
6	라
7	바
8	사
9	나

- 학생 테이블에서 CS 학과 3학년의 학번과 이름을 검색하라.

학번	이름	…	…	학과	…	학년
1	가					
2	나					
3	다					
4	라			CS		3
5	마			CS		3
6	라			CS		3
7	바					
8	사					
9	나					

- 방법 1: 셀렉트 + 프로젝트

학번	이름	…	…	학과	…	학년
4	라			CS		3
5	마			CS		3
6	라			CS		3

학번	이름
4	라
5	마
6	라

- 방법 2: 프로젝트 + 셀렉트
  - ◆ 셀렉트의 조건식을 적용할 수 없음 (학과와 학년이 없음).

학번	이름
1	가
2	나
3	다
4	라
5	마
6	라
7	바
8	사
9	나

조인 (JOIN, ⋈)

□ 세타조인 (theta-join)

- R(X), S(Y), A∈X, B∈Y 에 대하여 (A,B는 조인 애트리뷰트)

$$R \bowtie_{A\theta B} S = \{ r \cdot s \mid r \in R \wedge s \in S \wedge (\underline{r.A \theta s.B}) \}$$
$$= \sigma_{A\theta B} (R \times S)$$

- 결과 차수 = R의 차수 + S의 차수
- example
  - ◆ 학생 $\bowtie_{\text{학번=학번}}$ 등록

□ 동일조인 (equi-join)

- 세타조인에서 θ가 "="인 경우

$$R \bowtie_{A=B} S = \{ r \cdot s \mid r \in R \wedge s \in S \wedge (\underline{r.A = s.B}) \}$$

□ 자연조인 (natural join, $\bowtie_N$)

- R(X), S(Y)의 조인 애트리뷰트를 Z(=X∩Y)라 하면

$$R \bowtie_N S = \{ \underline{<r \cdot s>[X \cup Y]} \mid r \in R \wedge s \in S \wedge \underline{r[Z] = s[Z]} \}$$
$$= \Pi_{X \cup Y}(\sigma_{Z=Z}(R \times S))$$
$$= \Pi_{X \cup Y}(R \bowtie_{Z=Z} S)$$

- Note

  동일조인: $R \bowtie_{A=B} S = \{ \underline{<r \cdot s>} \mid r \in R \wedge s \in S \wedge \underline{r[Z] = s[Z]} \}$

  세타조인: $R \bowtie_{A \theta B} S = \{ \underline{<r \cdot s>} \mid r \in R \wedge s \in S \wedge \underline{r[Z] \theta s[Z]} \}$

- 즉 동일조인의 결과 릴레이션에서 애트리뷰트의 중복을 제거함.
- 결과 차수 = R의 차수 + S의 차수 − |X∩Y|

□ ☞ 동일조인 (equi-join) 처리 알고리즘

```
For each r∈R do
        For each s∈S do
                if r.A = s.B then print r · s;
        end;
end;
```

예제: Equi-Join 연산

□ 선수들의 이름, 백넘버, 소속 팀명 및 연고지를 검색하라.

선수(PLAYER) 테이블

PLAYER_NAME	BACK_NO	TEAM_ID
이 고 르	21	K06
오 비 나	26	K10
윤 원 일	45	K02
페 르 난 도	44	K04
레 오	45	K03
실 바	45	K07
무 스 타 파	77	K04
에 디	7	K01
알리송	14	K014
쟈스민	33	K08
디디	8	K06
⋮	⋮	⋮

팀(TEAM) 테이블

TEAM_ID	TEAM_NAME	REGION_NAME
K05	현대모터스	전부
K08	일화천마	성남
K03	스틸러스	포항
K07	드래곤즈	전남
K09	FC서울	서울
K04	유나이티드	인천
K11	경남FC	경남
K01	울산현대	울산
K10	시티즌	대전
K02	삼성블루윙즈	수우 js
K12	광주상무	광주
K06	아이파크	부산
K13	강원FC	강원
K14	제주유나이티드FC	재주
K15	대구FC	대구

선수(PLAYER) 테이블			팀(TEAM) 테이블		
PLAYER_NAME	BACK_NO	TEAM_ID	TEAM_ID	TEAM_NAME	REGION_NAME
이 고 르	21	K06	K06	아이파크	부산
오 비 나	26	K10	K10	시티즌	대전
윤 원 일	45	K02	K02	삼성블루윙즈	수원
페 르 난 도	44	K04	K04	유나이티드	인천
레 오	45	K03	K03	스틸러스	포항
실 바	45	K07	K07	드래곤즈	전남
무 스 타 파	77	K04	K04	유나이티드	인천
에 디	7	K01	K01	울산현대	울산
알리송	14	K014	K01	울산현대	울산
쟈스민	33	K08	K08	일화천마	성남
디디	8	K06	K06	아이파크	부산
⋮	⋮	⋮	⋮	⋮	⋮

디비전 (DIVISION, ÷)

□ 정의

- 릴레이션 R(X), S(Y) 에 대하여
  $Y \subseteq X$이고 $X-Y=Z$이면, R(X)=R(Z,Y)로 표현 가능.
- 따라서 R(Z,Y), S(Y) 에 대하여

$$R \div S = \{ t \mid \underline{t \in \Pi_{Z}(R)} \ \wedge \ \underline{t \cdot s \in R \text{ for all } s \in S} \}$$

◻ example

등록(E)

학번 (SNO)	과목번호 (CNO)
100	C413
100	E412
200	C123
300	C312
300	C324
300	C413
400	C312
400	C324
400	C413
400	E412
500	C312

과목1(C1)

과목번호 (CNO)
C413

과목2(C2)

과목번호 (CNO)
C312
C413

과목3(C3)

과목번호 (CNO)
C312
C413
E412

E ÷ C1

학번 (SNO)
100
300
400

E ÷ C2

학번 (SNO)
300
400

E ÷ C3

학번 (SNO)
400

작명 연산 (RENAME, ρ)

- 중간 결과 릴레이션에 이름을 붙이거나 애트리뷰트 이름을 변경할 때 사용

① $\rho_S(E)$

관계 대수식 E의 결과 릴레이션의 이름을 S로 지정

② $\rho_{S(B_1,B_2,\ \ldots\ ,B_m)}(E)$

관계 대수식 E의 결과 릴레이션의 이름을 S로 하면서 애트리뷰트 이름을 $B_1,B_2,\ \ldots\ ,B_m$ 으로 지정

9.1.3 근원 연산과 복합 연산

- 근원연산 (primitive operations)
  - 합집합, 차집합, 카티션 프로덕트, 프로젝트, 실렉트

- 복합연산 (composite operations)
  - 교집합, 조인, 디비전

- 복합연산은 근원연산으로 표현 가능
  - $R \cap S = R-(R-S) = S-(S-R)$
    $= (R \cup S)-((R-S) \cup (S-R))$
  - $R \bowtie_{A\theta B} S = \sigma_{A\theta B}(R \times S)$
  - $R(Z,Y) \div S(Y) = R[Z] - ((R[Z] \times S) - R)[Z]$

9.1.4 관계 대수의 확장

- 세미조인
- 외부조인
- 외부 합집합
- 집단 연산

세미조인 (SEMIJOIN, $\ltimes$)

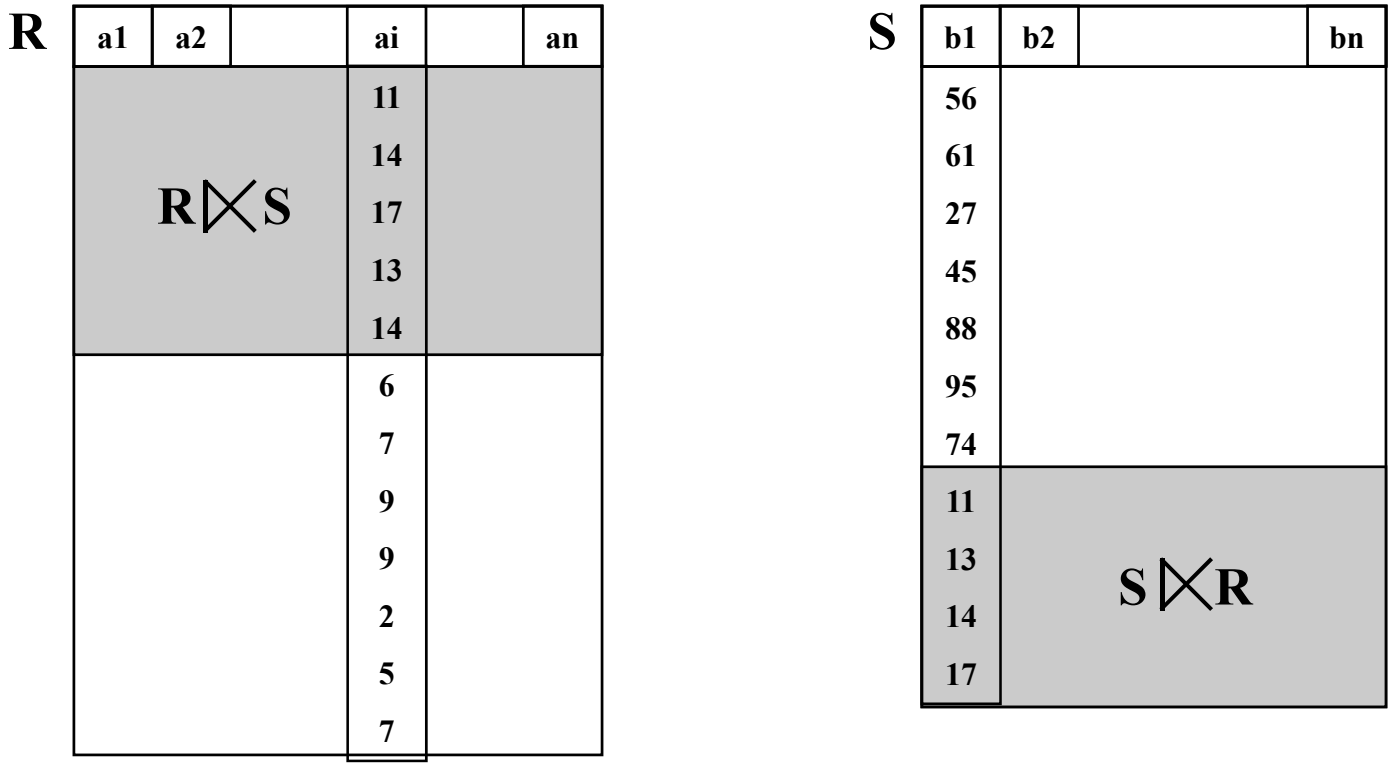
- 정의
  - R(X), S(Y)의 조인 애트리뷰트를 X∩Y라 하면,

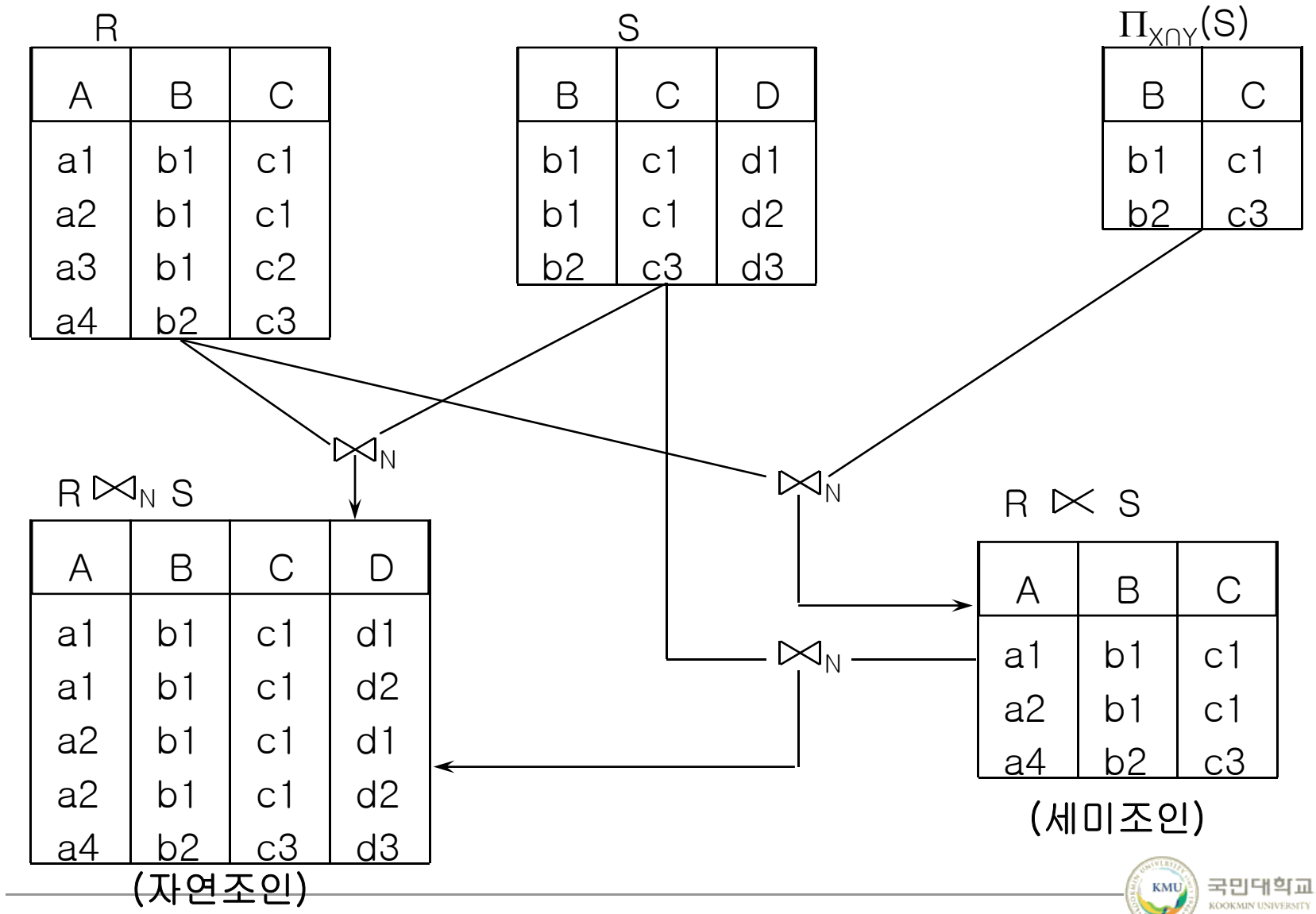
$$R \ltimes S = R \bowtie_N (\Pi_{X \cap Y}(S)) = \Pi_X (R \bowtie_N S)$$

- S와 자연조인을 할 수 있는 R의 투플

- 특징
  - $R \ltimes S \neq S \ltimes R$
  - $R \bowtie_N S = (R \ltimes S) \bowtie_N S = (S \ltimes R) \bowtie_N R$
    $= (R \ltimes S) \bowtie_N (S \ltimes R)$

R⋈S


R
a1
a2
ai
an
R⋉S
11
14
17
13
14
6
7
9
9
2
5
7
S
b1
b2
bn
56
61
27
45
88
95
74
11
13
14
17
S⋉R

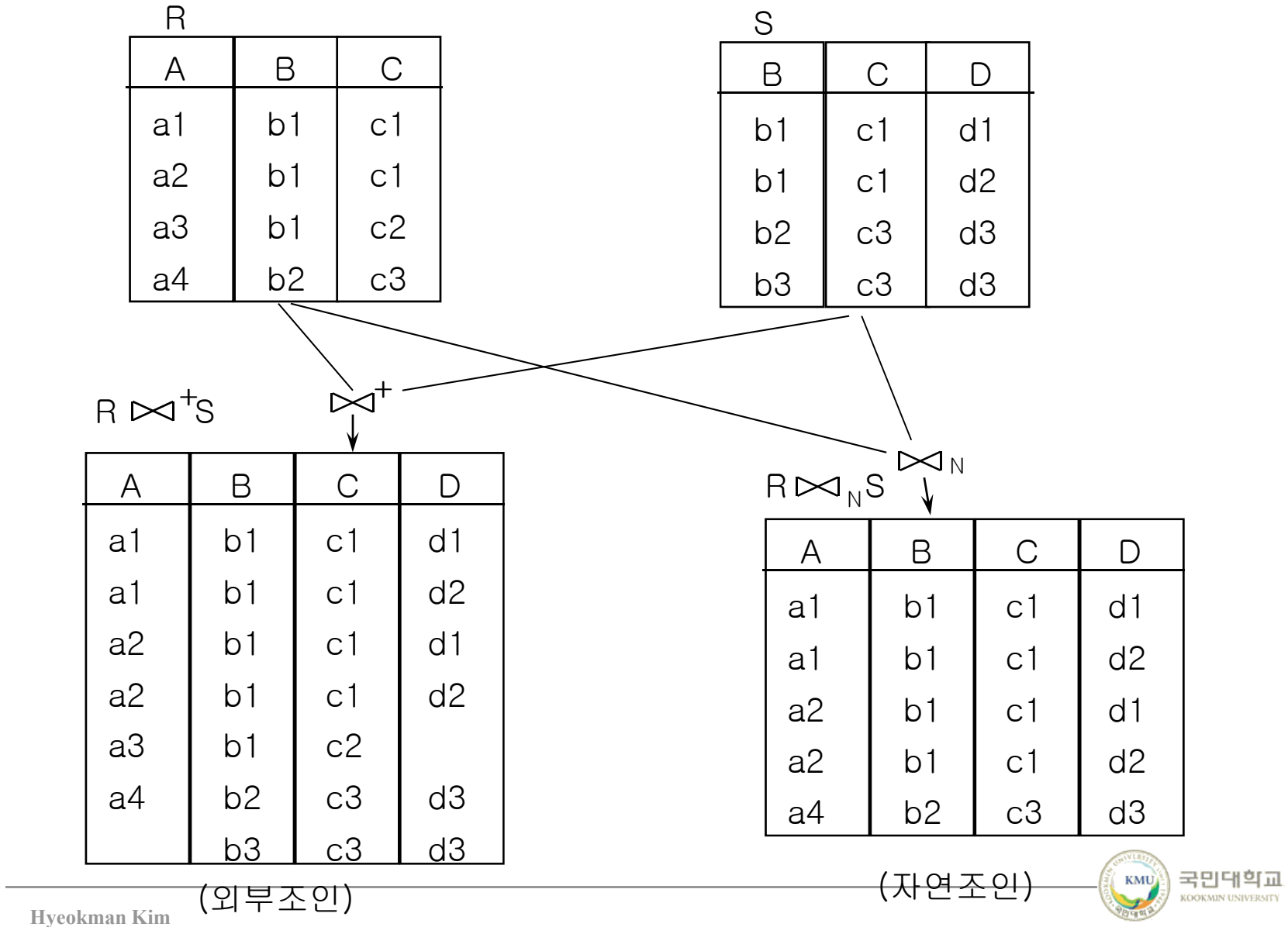

R
A B C
a1 b1 c1
a2 b1 c1
a3 b1 c2
a4 b2 c3
S
B C D
b1 c1 d1
b1 c1 d2
b2 c3 d3
$\Pi_{X\cap Y}(S)$
B C
b1 c1
b2 c3
$\bowtie_N$
$\bowtie_N$
$\bowtie_N$
R $\bowtie_N$ S
A B C D
a1 b1 c1 d1
a1 b1 c1 d2
a2 b1 c1 d1
a2 b1 c1 d2
a4 b2 c3 d3
(자연조인)
R $\ltimes$ S
A B C
a1 b1 c1
a2 b1 c1
a4 b2 c3
(세미조인)

- 분산 환경 (R: 서울, S: 서울)
  - $R \bowtie_N S = (R \ltimes S) \bowtie_N S = (R \bowtie_N (\Pi_{X \cap Y}(S))) \bowtie_N S$
    - 질의: 부산
  - $R \bowtie_N S = (S \ltimes R) \bowtie_N R$
    - 질의: 서울
  - $R \bowtie_N S = (R \ltimes S) \bowtie_N (S \ltimes R)$
    - 질의: 제3의 장소

외부조인 (OUTERJOIN, $\bowtie^{+}$)

- 정의
  - 조인시 한 릴레이션에 있는 투플이 조인할 상대 릴레이션에 대응되는 투플이 없을 경우, 상대를 널(null) 투플로 만들어 결과 릴레이션에 포함

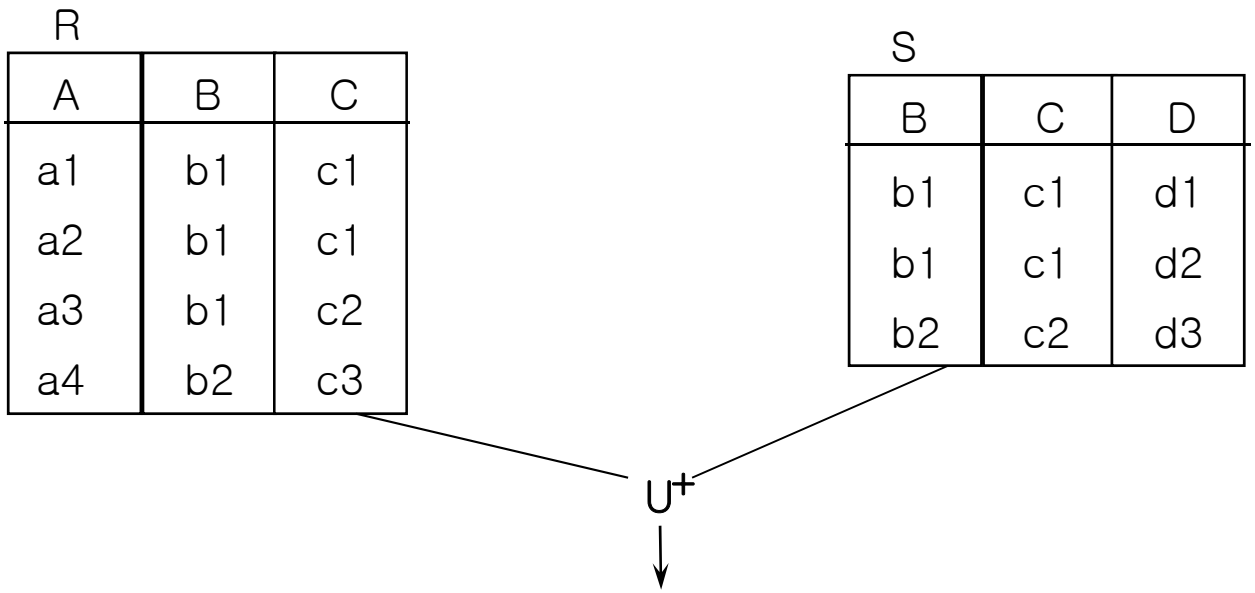
- 두 조인 릴레이션의 모든 투플들이 결과 릴레이션에 포함됨


R
A B C
a1 b1 c1
a2 b1 c1
a3 b1 c2
a4 b2 c3
S
B C D
b1 c1 d1
b1 c1 d2
b2 c3 d3
b3 c3 d3
R ⟗+ S
A B C D
a1 b1 c1 d1
a1 b1 c1 d2
a2 b1 c1 d1
a2 b1 c1 d2
a3 b1 c2
a4 b2 c3 d3
b3 c3 d3
(외부조인)
R ⟗N S
A B C D
a1 b1 c1 d1
a1 b1 c1 d2
a2 b1 c1 d1
a2 b1 c1 d2
a4 b2 c3 d3
(자연조인)

외부 합집합 (OUTER-UNION, $\cup^{+}$)

□ 정의

- 합병가능하지 않은(부분적으로 합병 가능한) 두 릴레이션을 차수를 확장시켜 합집합으로 만듬.



집단 연산

- 수학적 집단 연산
  - SUM, AVG, MAX, MIN, COUNT
  - $AVG_{성적}$(등록)
    - 등록 릴레이션에 있는 성적 애트리뷰트 값들에 대해 평균값 계산
  - 중복값이 있더라도 제외하지 않고 그대로 적용.
- 그룹 연산
  - $GROUP_{학년}$(학생)
    - 학생 릴레이션의 투플들을 학년 값에 따라 그룹 짓게 함
- 일반 형식
  - 일반형식: $F_B(G_A(E))$
    - E : 관계 대수식
    - G : 그룹연산
    - F : 집단 함수 (SUM, AVG, MAX, MIN, COUNT)
    - B : 집단 함수의 적용 대상 애트리뷰트
    - A : 그룹 함수가 적용할 애트리뷰트
  - 먼저 그룹 연산을 적용한후, 각 그룹에 대해 집단 함수를 적용.

Example: 대학(University) 관계 데이타베이스

학생
(STUDENT)

학번 (SNO)	이름 (SNANE)	학년 (YEAR)	학과 (DEPT)
100	나 연 묵	4	컴퓨터
200	이 찬 영	3	전기
300	정 기 태	1	컴퓨터
400	송 병 호	4	컴퓨터
500	박 종 화	2	산공

과목
(COURSE)

과목번호 (CNO)	과목이름 (CNANE)	학점 (CREDIT)	학과 (DEPT)	담당교수 (PRNAME)
C123	프로그래밍	3	컴퓨터	김성기
C312	자료 구조	3	컴퓨터	황수찬
C324	파일 처리	3	컴퓨터	이규철
C413	데이타 베이스	3	컴퓨터	이석호
C412	반 도 체	3	전자	홍봉희

등록
(ENROL)

학번 (SNO)	과목번호 (CNO)	성적 (GRADE)	중간성적 (MIDTERM)	기말성적 (FINAL)
100	C413	A	90	95
100	E412	A	95	95
200	C123	B	85	80
300	C312	A	90	95
300	C324	C	75	75
300	C413	A	95	90
400	C312	A	90	95
400	C324	A	95	90
400	C413	B	80	85
400	E412	C	65	75
500	C312	B	85	80

9.1.5 관계대수의 질의문 표현

- 모든 학생의 이름과 학과를 보여라.

- 모든 과목에 수강하고 있는 학생의 학번, 이름은?

- 과목번호가 C413인 과목을 등록하지 않은 학생의 이름은?

9.2 관계 해석 (Relational Calculus)

- Predicate calculus에 기반
- 관계 데이타 모델의 연산 표현 방법
- 비절차적(non-procedural)
  - 원하는 정보가 무엇이라는 것만 선언
- 종류
  - 투플 관계 해석 (tuple relational calculus)
  - 도메인 관계 해석 (domain relational calculus)

9.2.1 투플 관계해석

- 원하는 릴레이션을 투플해석식(tuple calculus expression)으로 명세

- 투플 해석식의 구성 요소

  i . 투플변수(tuple variable) 또는 범위변수(range variable): t
  - 범위식(range formula) : R(t)

  ii . 한정(qualified) 애트리뷰트 : t.A 또는 t[A]
  - 투플 변수 t가 나타내는 투플의 어떤 애트리뷰트 A의 값

iii. 원자(atom)

① R(t)

t : 투플변수

R : t의 범위 릴레이션

② t.AΘu.B

t, u : 투플변수

A, B : t와 u에 대한 한정 애트리뷰트

Θ : 비교 연산자(=, ≠, <, ≤, >,≥)

③ t.AΘc

A : 투플변수 t에 대한 한정 애트리뷰트

c : 상수

– 원자의 실행 결과는 반드시 참(True) 또는 거짓(False)

iv. 정형식(WFF, Well-formed formula)

- 원자, 불리언 연산자(∧, ∨, ¬), 정량자 (∀, ∃)가 다음 규칙에 따라 결합된 식
  - ① 모든 원자는 WFF
  - ② F가 WFF이면, (F)와 ¬F도 WFF
  - ③ F와 G가 WFF이면, F∧G와 F∨G도 WFF
  - ④ F가 WFF이고 t가 자유변수이면, ∀t(F)와 ∃t(F)도 WFF
  - ⑤ 위의 규칙만을 반복 적용해서 만들어진 식은 WFF

- 정형식의 예

  s.SNO = 100

  c.CNO≠e.CNO

  s.SNO = e.SNO ∧ e.CNO ≠ c.CNO

  (∃e)(e.SNO = s.SNO ∧ e.CNO = 'C413')

Note

- 속박변수(bound variable)
  - 정량자로 한정된 투플 변수
  - ∀ : 전칭 전량자(Universal quantifier)
  - ∃ : 존재 전량자(Existential quantifier)

- 자유변수(free variable)
  - 정량자로 한정되지 않는 투플 변수

투플 해석식 (Tuple calculus expression)

□ 형식

$\{ t_1.A_1, t_2.A_2, \ldots, t_n.A_n \mid F(t_1, \ldots t_n, t_{n+1}, \ldots, t_{n+m}) \}$

- t_i : 투플 변수
- $F(t_1,\ldots, t_n, t_{n+1},\ldots, t_{n+m})$: t_i가 연관된 정형식

- 막대 (|) 왼편에 나온 한정 애트리뷰트들은 목표 리스트로서, 막대(|) 오른편에 명세된 조건을 만족하는 결과로 추출 됨

□ example

{ s.SNAME | STUDENT(s) }

{ s.SNAME | STUDENT(s) ∧ s.DEPT='컴퓨터' }

투플 해석식의 질의문 표현

- 과목 C413에서 성적이 A인 모든 학생의 학번은?
  - { e.SNO | ENROL(e) ∧ e.CNO='C413' ∧ e.GRADE='A' }

- 과목 C413을 등록한 모든 학생의 이름과 학과는?
  - { s.SNAME, s.DEPT | STUDENT(s) ∧
    ∃e (ENROL(e) ∧ e.SNO=s.SNO ∧ e.CNO='C413') }

- 모든 과목에 등록한 학생의 이름을 전부 기술하라.
  - { s.SNAME | STUDENT(s) ∧
    (∃e) (∀c) (ENROL(e) ∧ COURSE(c) ∧ e.SNO=s.SNO
    ∧ e.CNO=c.CNO) }

- 과목 C413에 등록하지 않은 학생의 이름 전부를 기술하라.
  - {s.SNAME | STUDENT(s) ∧
    (¬∃e) (ENROL(e) ∧ e.SNO=s.SNO ∧ e.CNO='C413') }

9.2.3 도메인 관계해석

- 원하는 릴레이션을 도메인 해석식(domain calculus expression)으로 명세

- 도메인 해석식의 구성요소

  i . 도메인 변수(domain variable)
  - dSNO, dSNAME, ...
  - 범위식
    - STUDENT(dSNO, dSNAME, dDEPT, dYEAR)

ii . 원자(atom)

① $R(x_1,x_2,\ldots,x_n)$

x_i : 도메인 변수

R : x_i의 범위 릴레이션

$<x_1,x_2,\ldots x_n>$에 해당하는 값의 리스트는 릴레이션 R의 투플

② $x\theta y$

x, y : 도메인 변수

θ: 비교 연산자(=, ≠, <, ≤, >,≥)

③ $x\theta c$

x : 도메인 변수

θ: 비교 연산자

c : x가 정의된 도메인 값의 상수

- 원자의 실행 결과는 반드시 참(True) 또는 거짓(False)

iii. 정형식(WFF, Well-formed formula)

- 원자, 불리언 연산자(∧, ∨, ¬), 정량자(∀, ∃)가 다음 규칙에 따라 결합되어 표현된 식

① 모든 원자는 WFF
② F가 WFF이면, (F)와 ¬F도 WFF
③ F와 G가 WFF이면, F∧G와 F∨G도 WFF
④ F가 WFF이고 x가 자유변수이면, (∀x)(F)와 (∃x)(F)도 WFF
⑤ 위의 규칙만을 반복 적용해서 만들어진 식은 WFF

도메인 해석식(Domain calculus expression)

- 형식

  $\{ x_1, x_2, \ldots, x_n \mid F(x_1, \ldots x_n, x_{n+1}, \ldots, x_{n+m}) \}$

  - x_i : 도메인변수
  - $F(x_1, \ldots x_n, x_{n+1}, \ldots, x_{n+m})$: x_i에 대한 정형식
  - 막대 (|) 왼편에 나온 도메인 변수들은 목표리스트로서, 막대 (|) 오른편에 명세된 조건을 만족하는 도메인 값으로 만들어지는 투플

- Example
  - { dSNAME | STUDENT(dSNO, dSNAME, dYEAR, dDEPT)}
  - { dSNAME | (∃dDEPT) (STUDENT(dSNO, dSNAME, dYEAR, dDEPT)∧ dDEPT='컴퓨터') }
  - { dSNO, dDEPT | STUDENT(dSNO, dSNAME, dYEAR, dDEPT) ∧(∃ddSNO)(∃dGRADE)(ENROL(ddSNO, dCNO, dGRADE, dMIDTERM, dFINAL)∧dSNO=ddSNO∧dGRADE='A')

도메인 해석식의 질의문 표현

- 컴퓨터학과 3,4 학년의 이름을 보여라.
  - {dSNAME | (∃dYEAR)(∃dDEPT)(STUDENT(dSNO, dSNAME, dYEAR, dDEPT)∧ dYEAR ≥ 3∧ dDEPT='컴퓨터') }

- 과목 C413에서 성적이 A인 모든 학생의 학번은?
  - { dSNO | (∃dCNO)(∃dGRADE)(ENROL(dSNO, dCNO, dGRADE, dMIDTERM, dFINAL) ∧ dCNO= ‘C413’ ∧ dGRADE=‘A’) }

- 기말 성적이 90점 이상인 학생의 학번과 이름을 보여라.
  - {dSNO,dSNAME | (STUDENT(dSNO,dSNAME,dYEAR,dDEPT) ∧ (∃dFINAL)(∃ddSNO) (ENROL(ddSNO, dCNO, dGRADE, dMIDTERM, dFINAL)∧dSNO=ddSNO ∧ dFINAL ≥ 90) }

- 과목 C324에 등록하지 않은 학생의 이름은?
  - { dSNAME | (∃dSNO)((STUDENT(dSNAME, dSNO, dYEAR, dDEPT) ∧ (¬∃ddSNO) (∃dCNO) (ENROL(ddSNO, dCNO, dGRADE, dMIDTERM, dFINAL) ∧ dSNO=ddSNO ∧ dCNO='C324')) }

9.3 QBE

- QBE (Query By Example)
- 1975, IBM
- 도메인 관계 해석 사용
- 그래픽 디스플레이 단말기 사용
- 이차원 구문(two-dimensional syntax) 언어
- 테이블 형태
- 예(example)를 질의문 명세에 사용
  - 예제 원소(example element) : 도메인 변수

데이타 검색

□ 단순 조건 검색

- 4학년 학생의 학번과 이름을 보여라

STUDENT	SNO	SNAME	YEAR	DEPT
	P.	P.	4	컴퓨터

- 중복되는 것은 자동적으로 제거됨
- 'ALL'을 삽입하면 중복이 가능

□ 테이블 전체의 검색

STUDENT	SNO	SNAME	YEAR	DEPT
	P.	P.	P.	P.

□ 복수 조건 검색

- 'OR' 조건 : 두 개의 행, 다른 예제 원소
  - ◆ 기말성적이 85점 이상이거나 과목번호 ‘C413’에 등록한 학생의 학번을 검색하라

ENROL	SNO	CNO	FINAL	MIDTERM
	P._STX P._STY	C413	≥85	

- 'AND' 조건 : 하나의 행, 같은 예제 원소
  - ◆ 과목번호가 ‘C413’이고 기말성적이 85점 이상인 학생의 학번

ENROL	SNO	CNO	FINAL	MIDTERM
	P.	C413	≥85	

– 조건 상자(condition box)의 사용

ENROL	SNO	CNO	FINAL	MIDTERM
	P.	_EC	_EF	

CONDITIONS
_EC=C413 AND _EF ≥85

◘ 복수 테이블에서 검색

– 기말성적이 85점 이상이거나 과목 ‘C413’을 등록한 학생의 이름

ENROL	SNO	CNO	FINAL	MIDTERM
	_STX _STY	C413	≥85	

STUDENT	SNO	SNAME	YEAR	DEPT
	_STX _STY	P. P.		

데이타의 삽입

□ 단순 레코드의 삽입

- 학번이 100이고 과목번호가 'C413'인 투플 삽입

ENROL	SNO	CNO	GRADE	MIDTERM	FINAL
I.	100	C413			

□ 투플 검색을 이용한 삽입

- 4학년 학생의 학번을 학생테이블로부터 검색해서 SENIOR 테이블에 삽입하라.

SENIOR	SNO
I.	_STX

STUDENT	SNO	SNAME	YEAR	DEPT
	_STX		4	

데이타의 삭제

◘ 한 테이블에서의 삭제

– 학번이 100인 학생을 학생 테이블에서 삭제

STUDENT	SNO	SNAME	YEAR	DEPT
D.	100			

◘ 복수 테이블에서의 레코드 삭제

– 기말성적이 60점 미만인 학생을 등록 테이블과 학생테이블에서 삭제

ENROL	SNO	CNO	GRADE	MIDTERM	FINAL
D.	_STX _STX				<60

STUDENT	SNO	SNAME	YEAR	DEPT
D.	_STX			

데이타의 갱신

- 필드값의 단순 갱신
  - 학번이 300인 학생의 학년을 2로 변경

STUDENT	SNO	SNAME	YEAR	DEPT
	300		U.2	

STUDENT	SNO	SNAME	YEAR	DEPT
U.	300		2	

- 산술식을 이용한 갱신
  - 과목 'C413'에 등록한 학생의 기말 성적(FINAL)에 5점을 가산

ENROL	SNO	CNO	FINAL	MIDTERM
U.		C413	_G+5 _G	