

SQL

Suan Lee



Introduction to SQL

- “S.Q.L.” or “sequel”
- Supported by all major commercial database systems
- Standardized – many new features over time
- Interactive via GUI or prompt, or embedded in programs
- Declarative, based on relational algebra

Data Definition Language (DDL)

- CREATE TABLE ...
- DROP TABLE ...

Data Manipulation Language (DML)

- SELECT ...
- INSERT ...
- DELETE ...
- UPDATE ...

Other Commands

indexes, constraints, views, triggers, transactions, authorization, ...

The Basic SELECT Statement

SELECT	A1, A2, ..., An
FROM	R1, R2, ..., Rm
WHERE	condition

The Basic SELECT Statement

SELECT	$A_1, A_2, \dots, A_n$
FROM	$R_1, R_2, \dots, R_m$
WHERE	condition

$$\Pi_{A_1, \dots, A_n}(\sigma_{\text{condition}}(R_1 \times \dots \times R_m))$$

- “S.Q.L.” or “sequel”
- Supported by all major commercial database systems
- Standardized – many new features over time
- Interactive via GUI or prompt, or embedded in programs
- Declarative, based on relational algebra

Basic SELECT Statement

SELECT	$A_1, A_2, \dots, A_n$
FROM	$R_1, R_2, \dots, R_m$
WHERE	condition

Basic SELECT Statement

SELECT	$A_1, A_2, \dots, A_n$
FROM	$R_1, R_2, \dots, R_m$
WHERE	condition

Demo: simple college admissions database

College(cName, state, enrollment)

Student(sID, sName, GPA, sizeHS)

Apply(sID, cName, major, decision)

- Colleges

	A Z CNAME	A Z STATE	A Z ENROLLMENT
1	Stanford	CA	15000
2	Berkeley	CA	36000
3	MIT	MA	10000
4	Cornell	NY	21000

- Students

	AZ	SID	AZ	SNAME	AZ	GPA	AZ	SIZEHS
1		123		Amy		3.9		1000
2		234		Bob		3.6		1500
3		345		Craig		3.5		500
4		456		Doris		3.9		1000
5		567		Edward		2.9		2000
6		678		Fay		3.8		200
7		789		Gary		3.4		800
8		987		Helen		3.7		800
9		876		Irene		3.9		400
10		765		Jay		2.9		1500
11		654		Amy		3.9		1000
12		543		Craig		3.4		2000

- Apply

	RZ SID	RZ CNAME	RZ MAJOR	RZ DECISION
1	123	Stanford	CS	Y
2	123	Stanford	EE	N
3	123	Berkeley	CS	Y
4	123	Cornell	EE	Y
5	234	Berkeley	biology	N
6	345	MIT	bioengineering	Y
7	345	Cornell	bioengineering	N
8	345	Cornell	CS	Y
9	345	Cornell	EE	N
10	678	Stanford	history	Y
11	987	Stanford	CS	Y
12	987	Berkeley	CS	Y
13	876	Stanford	CS	N
14	876	MIT	biology	Y
15	876	MIT	marine biology	N
16	765	Stanford	history	Y
17	765	Cornell	history	N
18	765	Cornell	psychology	Y
19	543	MIT	CS	N

SQL

```
SELECT  sID, sName, GPA
FROM    Student
WHERE   GPA > 3.6;
```

Result

	R Z	SID	R Z	SNAME	R Z	GPA
1		123		Amy		3.9
2		456		Doris		3.9
3		678		Fay		3.8
4		987		Helen		3.7
5		876		Irene		3.9
6		654		Amy		3.9

SQL

```
SELECT  sID, sName  
FROM    Student  
WHERE   GPA > 3.6;
```

Result

	R Z	SID	R Z	SNAME
1		123		Amy
2		456		Doris
3		678		Fay
4		987		Helen
5		876		Irene
6		654		Amy

SQL

```
SELECT sName, major
FROM Student, Apply
WHERE Student.sID = Apply.sID;
```

Result

	 SNAME	 MAJOR
1	Amy	CS
2	Amy	EE
3	Amy	CS
4	Amy	EE
5	Bob	biology
6	Craig	bioengineering
7	Craig	bioengineering
8	Craig	CS
9	Craig	EE
10	Fay	history
11	Helen	CS
12	Helen	CS
13	Irene	CS
14	Irene	biology
15	Irene	marine biology
16	Jay	history
17	Jay	history
18	Jay	psychology
19	Craig	CS

SQL

```
SELECT sName, major
FROM Student, Apply
WHERE Student.sID = Apply.sID;
```

Result

	 SNAME	 MAJOR
1	Amy	CS
2	Amy	EE
3	Amy	CS
4	Amy	EE
5	Bob	biology
6	Craig	bioengineering
7	Craig	bioengineering
8	Craig	CS
9	Craig	EE
10	Fay	history
11	Helen	CS
12	Helen	CS
13	Irene	CS
14	Irene	biology
15	Irene	marine biology
16	Jay	history
17	Jay	history
18	Jay	psychology
19	Craig	CS

duplicate values
duplicate values

duplicate values

duplicate values

duplicate values

SQL

```
SELECT DISTINCT sName, major
FROM Student, Apply
WHERE Student.sID = Apply.sID;
```

Result

	R2 SNAME	R2 MAJOR
1	Amy	EE
2	Bob	biology
3	Craig	EE
4	Irene	biology
5	Jay	psychology
6	Amy	CS
7	Craig	CS
8	Jay	history
9	Craig	bioengineering
10	Fay	history
11	Irene	marine biology
12	Helen	CS
13	Irene	CS

SQL

```
SELECT sName, GPA, decision
FROM Student, Apply
WHERE Student.sID = Apply.sID
      AND sizeHS < 1000
      AND major = 'CS'
      AND cName = 'Stanford';
```

Result

	R Z SNAME	R Z GPA	R Z DECISION
1	Helen	3.7	Y
2	Irene	3.9	N

SQL

```
SELECT  cName
FROM    College, Apply
WHERE   College.cName =
        Apply.cName
        AND enrollment > 20000
        AND major = 'CS';
```

SQL

```
SELECT  cName
FROM    College, Apply
WHERE   College.cName =
        Apply.cName
        AND enrollment > 20000
        AND major = 'CS';
```

Result

```
ORA-00918: 열의 정의가 애매합니다
00918. 00000 - "column ambiguously defined"
*Cause:
*Action:
94행, 9열에서 오류 발생
```

SQL

```
SELECT College.cName
FROM College, Apply
WHERE College.cName =
        Apply.cName
        AND enrollment > 20000
        AND major = 'CS';
```

Result

	R#	CNAME
1	Berkeley	
2	Cornell	
3	Berkeley	

SQL

```
SELECT DISTINCT College.cName
FROM   College, Apply
WHERE  College.cName =
        Apply.cName
        AND enrollment > 20000
        AND major = 'CS';
```

Result

	R#	CNAME
	1	Cornell
	2	Berkeley

SQL

```
SELECT Student.sID, sName, GPA,  
        Apply.cName, enrollment  
FROM Student, College, Apply  
WHERE Apply.sID = Student.sID  
      AND Apply.cName =  
           College.cName;
```


SQL

```
SELECT Student.sID, sName, GPA,
        Apply.cName, enrollment
FROM Student, College, Apply
WHERE Apply.sID = Student.sID
      AND Apply.cName =
        College.cName;
```

Result

	A2	SID	A2	SNAME	A2	GPA	A2	CNAME	A2	ENROLLMENT
1		123	Amy			3.9		Cornell		21000
2		123	Amy			3.9		Berkeley		36000
3		123	Amy			3.9		Stanford		15000
4		123	Amy			3.9		Stanford		15000
5		234	Bob			3.6		Berkeley		36000
6		345	Craig			3.5		Cornell		21000
7		345	Craig			3.5		Cornell		21000
8		345	Craig			3.5		Cornell		21000
9		345	Craig			3.5		MIT		10000
10		678	Fay			3.8		Stanford		15000
11		987	Helen			3.7		Berkeley		36000
12		987	Helen			3.7		Stanford		15000
13		876	Irene			3.9		MIT		10000
14		876	Irene			3.9		MIT		10000
15		876	Irene			3.9		Stanford		15000
16		765	Jay			2.9		Cornell		21000
17		765	Jay			2.9		Cornell		21000
18		765	Jay			2.9		Stanford		15000
19		543	Craig			3.4		MIT		10000

SQL

```
SELECT Student.sID, sName, GPA,  
        Apply.cName, enrollment  
FROM Student, College, Apply  
WHERE Apply.sID = Student.sID  
      AND Apply.cName =  
           College.cName  
ORDER BY GPA DESC;
```

Result

	A2	SID	A2	SNAME	A2	GPA	A2	CNAME	A2	ENROLLMENT
1		123	Amy			3.9		Cornell		21000
2		123	Amy			3.9		Berkeley		36000
3		123	Amy			3.9		Stanford		15000
4		123	Amy			3.9		Stanford		15000
5		876	Irene			3.9		Stanford		15000
6		876	Irene			3.9		MIT		10000
7		876	Irene			3.9		MIT		10000
8		678	Fay			3.8		Stanford		15000
9		987	Helen			3.7		Berkeley		36000
10		987	Helen			3.7		Stanford		15000
11		234	Bob			3.6		Berkeley		36000
12		345	Craig			3.5		Cornell		21000
13		345	Craig			3.5		Cornell		21000
14		345	Craig			3.5		MIT		10000
15		345	Craig			3.5		Cornell		21000
16		543	Craig			3.4		MIT		10000
17		765	Jay			2.9		Cornell		21000
18		765	Jay			2.9		Cornell		21000
19		765	Jay			2.9		Stanford		15000

SQL

```
SELECT Student.sID, sName, GPA,
        Apply.cName, enrollment
FROM Student, College, Apply
WHERE Apply.sID = Student.sID
      AND Apply.cName =
          College.cName
ORDER BY GPA DESC, enrollment;
```

Result

	A2	SID	A2	SNAME	A2	GPA	A2	CNAME	A2	ENROLLMENT
1		876	Irene			3.9		MIT		10000
2		876	Irene			3.9		MIT		10000
3		123	Amy			3.9		Stanford		15000
4		123	Amy			3.9		Stanford		15000
5		876	Irene			3.9		Stanford		15000
6		123	Amy			3.9		Cornell		21000
7		123	Amy			3.9		Berkeley		36000
8		678	Fay			3.8		Stanford		15000
9		987	Helen			3.7		Stanford		15000
10		987	Helen			3.7		Berkeley		36000
11		234	Bob			3.6		Berkeley		36000
12		345	Craig			3.5		MIT		10000
13		345	Craig			3.5		Cornell		21000
14		345	Craig			3.5		Cornell		21000
15		345	Craig			3.5		Cornell		21000
16		543	Craig			3.4		MIT		10000
17		765	Jay			2.9		Stanford		15000
18		765	Jay			2.9		Cornell		21000
19		765	Jay			2.9		Cornell		21000

SQL

```
SELECT      SID, major
FROM        Apply
WHERE       major LIKE '%bio%';
```

Result

	RZ	SID	RZ	MAJOR
1		234		biology
2		345		bioengineering
3		345		bioengineering
4		876		biology
5		876		marine biology

SQL

```
SELECT *
FROM Apply
WHERE major LIKE '%bio%';
```

Result

	SID	CNAME	MAJOR	DECISION
1	234	Berkeley	biology	N
2	345	MIT	bioengineering	Y
3	345	Cornell	bioengineering	N
4	876	MIT	biology	Y
5	876	MIT	marine biology	N

SQL

```
SELECT *  
FROM Student, College;
```

SQL

```
SELECT *  
FROM Student, College;
```

Result

	A2	SID	A2	SNAME	A2	GPA	A2	SIZEHS	A2	CNAME	A2	STATE	A2	ENROLLMENT
1		123		Amy		3.9		1000		Stanford		CA		15000
2		234		Bob		3.6		1500		Stanford		CA		15000
3		345		Craig		3.5		500		Stanford		CA		15000
4		456		Doris		3.9		1000		Stanford		CA		15000
5		567		Edward		2.9		2000		Stanford		CA		15000
6		678		Fay		3.8		200		Stanford		CA		15000
7		789		Gary		3.4		800		Stanford		CA		15000
8		987		Helen		3.7		800		Stanford		CA		15000
9		876		Irene		3.9		400		Stanford		CA		15000
10		765		Jay		2.9		1500		Stanford		CA		15000
11		654		Amy		3.9		1000		Stanford		CA		15000
12		543		Craig		3.4		2000		Stanford		CA		15000
13		123		Amy		3.9		1000		Berkeley		CA		36000
14		234		Bob		3.6		1500		Berkeley		CA		36000
15		345		Craig		3.5		500		Berkeley		CA		36000
16		456		Doris		3.9		1000		Berkeley		CA		36000
17		567		Edward		2.9		2000		Berkeley		CA		36000
										Berkeley		CA		36000

•
•
•

SQL

```
SELECT    sID, sName, GPA,  
          sizeHS,  
          GPA*(sizeHS/1000.0)  
FROM      Student;
```


SQL

```
SELECT    SID, sName, GPA,  
          sizeHS,  
          GPA*(sizeHS/1000.0)  
FROM      Student;
```

Result

	RZ	SID	RZ	SNAME	RZ	GPA	RZ	SIZEHS	RZ	GPA*(SIZEHS/1000.0)
1		123		Amy		3.9		1000		3.9
2		234		Bob		3.6		1500		5.4
3		345		Craig		3.5		500		1.75
4		456		Doris		3.9		1000		3.9
5		567		Edward		2.9		2000		5.8
6		678		Fay		3.8		200		0.76
7		789		Gary		3.4		800		2.72
8		987		Helen		3.7		800		2.96
9		876		Irene		3.9		400		1.56
10		765		Jay		2.9		1500		4.35
11		654		Amy		3.9		1000		3.9
12		543		Craig		3.4		2000		6.8

SQL

```
SELECT    SID, sName, GPA,  
          sizeHS,  
          GPA*(sizeHS/1000.0)  
          AS scaledGPA  
FROM      Student;
```

Result

	PK	SID	PK	SNAME	PK	GPA	PK	SIZEHS	PK	SCALEDGPA
1		123		Amy		3.9		1000		3.9
2		234		Bob		3.6		1500		5.4
3		345		Craig		3.5		500		1.75
4		456		Doris		3.9		1000		3.9
5		567		Edward		2.9		2000		5.8
6		678		Fay		3.8		200		0.76
7		789		Gary		3.4		800		2.72
8		987		Helen		3.7		800		2.96
9		876		Irene		3.9		400		1.56
10		765		Jay		2.9		1500		4.35
11		654		Amy		3.9		1000		3.9
12		543		Craig		3.4		2000		6.8

Table Variables and Set Operators

SELECT
FROM
WHERE

$A_1, A_2, \dots, A_n$
 $R_1, R_2, \dots, R_m$
condition

Table variables

Set Operators: Union, Intersect, Except

Demo: simple college admissions database

College(cName, state, enrollment)

Student(sID, sName, GPA, sizeHS)

Apply(sID, cName, major, decision)

SQL

```
SELECT      Student.sID, sName,  
            GPA, Apply.cName,  
            enrollment  
FROM        Student, College,  
            Apply  
WHERE       Apply.sID =  
            Student.sID  
            AND  
            Apply.cName =  
            College.cName;
```

Result

	AZ	SID	AZ	SNAME	AZ	GPA	AZ	CNAME	AZ	ENROLLMENT
1		123		Amy		3.9		Cornell		21000
2		123		Amy		3.9		Berkeley		36000
3		123		Amy		3.9		Stanford		15000
4		123		Amy		3.9		Stanford		15000
5		234		Bob		3.6		Berkeley		36000
6		345		Craig		3.5		Cornell		21000
7		345		Craig		3.5		Cornell		21000
8		345		Craig		3.5		Cornell		21000
9		345		Craig		3.5		MIT		10000
10		678		Fay		3.8		Stanford		15000
11		987		Helen		3.7		Berkeley		36000
12		987		Helen		3.7		Stanford		15000
13		876		Irene		3.9		MIT		10000
14		876		Irene		3.9		MIT		10000
15		876		Irene		3.9		Stanford		15000
16		765		Jay		2.9		Cornell		21000
17		765		Jay		2.9		Cornell		21000
18		765		Jay		2.9		Stanford		15000
19		543		Craig		3.4		MIT		10000

SQL

```
SELECT      S.sID, sName,  
            GPA, A.cName,  
            enrollment  
FROM        Student S, College C,  
            Apply A  
WHERE       A.sID = S.sID  
AND         A.cName = C.cName;
```

Result

	AZ	SID	AZ	SNAME	AZ	GPA	AZ	CNAME	AZ	ENROLLMENT
1		123	Amy		3.9	Cornell		21000		
2		123	Amy		3.9	Berkeley		36000		
3		123	Amy		3.9	Stanford		15000		
4		123	Amy		3.9	Stanford		15000		
5		234	Bob		3.6	Berkeley		36000		
6		345	Craig		3.5	Cornell		21000		
7		345	Craig		3.5	Cornell		21000		
8		345	Craig		3.5	Cornell		21000		
9		345	Craig		3.5	MIT		10000		
10		678	Fay		3.8	Stanford		15000		
11		987	Helen		3.7	Berkeley		36000		
12		987	Helen		3.7	Stanford		15000		
13		876	Irene		3.9	MIT		10000		
14		876	Irene		3.9	MIT		10000		
15		876	Irene		3.9	Stanford		15000		
16		765	Jay		2.9	Cornell		21000		
17		765	Jay		2.9	Cornell		21000		
18		765	Jay		2.9	Stanford		15000		
19		543	Craig		3.4	MIT		10000		

SQL

```
SELECT      S1.sID, S1.sName,  
            S1.GPA,  
            S2.sID, S2.sName,  
            S2.GPA  
FROM        Student s1, Student s2  
WHERE       S1.GPA = S2.GPA;
```

SQL

```
SELECT      S1.sID, S1.sName,  
            S1.GPA,  
            S2.sID, S2.sName,  
            S2.GPA  
FROM        Student S1, Student S2  
WHERE       S1.GPA = S2.GPA;
```

Result

	RZ	SID	RZ	SNAME	RZ	GPA	RZ	SID_1	RZ	SNAME_1	RZ	GPA_1
1		654		Amy		3.9		123		Amy		3.9
2		876		Irene		3.9		123		Amy		3.9
3		456		Doris		3.9		123		Amy		3.9
4		123		Amy		3.9		123		Amy		3.9
5		234		Bob		3.6		234		Bob		3.6
6		345		Craig		3.5		345		Craig		3.5
7		654		Amy		3.9		456		Doris		3.9
8		876		Irene		3.9		456		Doris		3.9
9		456		Doris		3.9		456		Doris		3.9
10		123		Amy		3.9		456		Doris		3.9
11		765		Jay		2.9		567		Edward		2.9
12		567		Edward		2.9		567		Edward		2.9
13		678		Fay		3.8		678		Fay		3.8
14		543		Craig		3.4		789		Gary		3.4
15		789		Gary		3.4		789		Gary		3.4
16		987		Helen		3.7		987		Helen		3.7

•
•
•

SQL

```
SELECT      S1.sID, S1.sName,  
            S1.GPA,  
            S2.sID, S2.sName,  
            S2.GPA  
FROM        Student S1, Student S2  
WHERE       S1.GPA = S2.GPA  
AND        S1.sID <> S2.sID;
```

Result

	RZ	SID	RZ	SNAME	RZ	GPA	RZ	SID_1	RZ	SNAME_1	RZ	GPA_1
1		654		Amy		3.9		123		Amy		3.9
2		876		Irene		3.9		123		Amy		3.9
3		456		Doris		3.9		123		Amy		3.9
4		654		Amy		3.9		456		Doris		3.9
5		876		Irene		3.9		456		Doris		3.9
6		123		Amy		3.9		456		Doris		3.9
7		765		Jay		2.9		567		Edward		2.9
8		543		Craig		3.4		789		Gary		3.4
9		654		Amy		3.9		876		Irene		3.9
10		456		Doris		3.9		876		Irene		3.9
11		123		Amy		3.9		876		Irene		3.9
12		567		Edward		2.9		765		Jay		2.9
13		876		Irene		3.9		654		Amy		3.9
14		456		Doris		3.9		654		Amy		3.9
15		123		Amy		3.9		654		Amy		3.9
16		789		Gary		3.4		543		Craig		3.4

SQL

```
SELECT      S1.sID, S1.sName,  
            S1.GPA,  
            S2.sID, S2.sName,  
            S2.GPA  
FROM        Student S1, Student S2  
WHERE       S1.GPA = S2.GPA  
AND        S1.sID < S2.sID;
```

Result

	PK SID	PK SNAME	PK GPA	PK SID_1	PK SNAME_1	PK GPA_1
1	123	Amy	3.9	456	Doris	3.9
2	543	Craig	3.4	789	Gary	3.4
3	654	Amy	3.9	876	Irene	3.9
4	456	Doris	3.9	876	Irene	3.9
5	123	Amy	3.9	876	Irene	3.9
6	567	Edward	2.9	765	Jay	2.9
7	456	Doris	3.9	654	Amy	3.9
8	123	Amy	3.9	654	Amy	3.9

SQL

```
SELECT      cName
FROM        College
UNION
SELECT      sName
FROM        Student;
```

Result

	R/C	CNAME
1		Amy
2		Berkeley
3		Bob
4		Cornell
5		Craig
6		Doris
7		Edward
8		Fay
9		Gary
10		Helen
11		Irene
12		Jay
13		MIT
14		Stanford

SQL

```
SELECT      cName AS name
FROM        College
UNION
SELECT      sName AS name
FROM        Student;
```

Result

	NAME
1	Amy
2	Berkeley
3	Bob
4	Cornell
5	Craig
6	Doris
7	Edward
8	Fay
9	Gary
10	Helen
11	Irene
12	Jay
13	MIT
14	Stanford

SQL

```
SELECT      cName AS name
FROM        College
UNION ALL
SELECT      sName AS name
FROM        Student;
```

Result

	NAME
1	Stanford
2	Berkeley
3	MIT
4	Cornell
5	Amy
6	Bob
7	Craig
8	Doris
9	Edward
10	Fay
11	Gary
12	Helen
13	Irene
14	Jay
15	Amy
16	Craig

SQL

```
SELECT      cName AS name
FROM        College
UNION ALL
SELECT      sName AS name
FROM        Student
ORDER BY    name;
```

Result

	NAME
1	Amy
2	Amy
3	Berkeley
4	Bob
5	Cornell
6	Craig
7	Craig
8	Doris
9	Edward
10	Fay
11	Gary
12	Helen
13	Irene
14	Jay
15	MIT
16	Stanford

SQL

```
SELECT      sID
FROM        Apply
WHERE       major = 'CS'
INTERSECT
SELECT      sID
FROM        Apply
WHERE       major = 'EE';
```

Result

	RZ	SID
1		123
2		345

SQL

```
SELECT  A1.sID
FROM    Apply A1, Apply A2
WHERE   A1.sID = A2.sID
        AND A1.major = 'CS'
        AND A2.major = 'EE';
```

Result

	A2	SID
1		123
2		123
3		123
4		123
5		345

SQL

```
SELECT  A1.sID
FROM    Apply A1, Apply A2
WHERE   A1.sID = A2.sID
        AND A1.major = 'CS'
        AND A2.major = 'EE';
```

Result

	A2	SID
1		123
2		123
3		123
4		123
5		345

	A2	SID	A2	CNAME	A2	MAJOR	A2	DECISION
1		123	Stanford	CS			Y	
2		123	Stanford	EE			N	
3		123	Berkeley	CS			Y	
4		123	Cornell	EE			Y	
5		234	Berkeley	biology			N	
6		345	MIT	bioengineering			Y	
7		345	Cornell	bioengineering			N	
8		345	Cornell	CS			Y	
9		345	Cornell	EE			N	
10		678	Stanford	history			Y	
11		987	Stanford	CS			Y	
12		987	Berkeley	CS			Y	
13		876	Stanford	CS			N	
14		876	MIT	biology			Y	
15		876	MIT	marine biology			N	
16		765	Stanford	history			Y	
17		765	Cornell	history			N	
18		765	Cornell	psychology			Y	
19		543	MIT	CS			N	

SQL

```
SELECT DISTINCT A1.sID
FROM      Apply A1, Apply A2
WHERE     A1.sID = A2.sID
        AND A1.major = 'CS'
        AND A2.major = 'EE';
```

Result

	A2	SID
1		123
2		345

SQL

```
SELECT      sID
FROM        Apply
WHERE       major = 'CS'
MINUS
SELECT      sID
FROM        Apply
WHERE       major = 'EE';
```

SQL

```
SELECT      sID
FROM        Apply
WHERE       major = 'CS'
MINUS
SELECT      sID
FROM        Apply
WHERE       major = 'EE';
```

Result

	RZ	SID
1		543
2		876
3		987

SQL

```
SELECT    A1.sID  
FROM      Apply A1, Apply A2  
WHERE     A1.sID = A2.sID  
    AND   A1.major = 'CS'  
    AND   A2.major <> 'EE';
```

SQL

```
SELECT      A1.sID
FROM        Apply A1, Apply A2
WHERE       A1.sID = A2.sID
           AND A1.major = 'CS'
           AND A2.major <> 'EE';
```

Result

	A2	SID
1		123
2		123
3		123
4		123
5		345
6		345
7		345
8		987
9		987
10		987
11		987
12		876
13		876
14		876
15		543

SQL

```
SELECT DISTINCT A1.sID
FROM      Apply A1, Apply A2
WHERE     A1.sID = A2.sID
        AND A1.major = 'CS'
        AND A2.major <> 'EE';
```

Result

	A2	SID
1		123
2		543
3		987
4		876
5		345

Subqueries in WHERE

SELECT
FROM
WHERE

$A_1, A_2, \dots, A_n$
 $R_1, R_2, \dots, R_m$
condition

Expressions involving
“subqueries”: nested
Select statements

Demo: simple college admissions database

College(cName, state, enrollment)

Student(sID, sName, GPA, sizeHS)

Apply(sID, cName, major, decision)

SQL

```
SELECT      sID, sName  
FROM        Student  
WHERE       sID IN  
(  
    SELECT      sID  
    FROM        Apply  
    WHERE       major = 'CS'  
);
```

SQL

```
SELECT      sID, sName
FROM        Student
WHERE       sID IN
(
    SELECT   sID
    FROM      Apply
    WHERE     major = 'CS'
);
```

Result

	R2	SID	R2	SNAME
1		123		Amy
2		345		Craig
3		987		Helen
4		876		Irene
5		543		Craig

SQL

```
SELECT    sID, sName  
FROM      Student, Apply  
WHERE     Student.sID =  
          Apply.sID  
AND       major = 'CS';
```

SQL

```
SELECT      sID, sName  
FROM        Student, Apply  
WHERE       Student.sID =  
            Apply.sID  
            AND  
            major = 'CS';
```

Result

```
ORA-00918: 열의 정의가 애매합니다  
00918. 00000 - "column ambiguously defined"  
*Cause:  
*Action:  
256행, 11열에서 오류 발생
```

SQL

```
SELECT      Student.sID, sName
FROM        Student, Apply
WHERE       Student.sID =
            Apply.sID
            AND
            major = 'CS';
```

Result

	R2	SID	R2	SNAME
1		123		Amy
2		123		Amy
3		345		Craig
4		987		Helen
5		987		Helen
6		876		Irene
7		543		Craig

SQL

```
SELECT      Student.sID, sName
FROM        Student, Apply
WHERE       Student.sID =
            Apply.sID
            AND
            major = 'CS';
```

```
SELECT      sID, sName
FROM        Student
WHERE       sID IN
(
    SELECT   sID
    FROM     Apply
    WHERE    major = 'CS'
);
```

Result

	R 2	SID	R 2	SNAME
1		123	Amy	
2		123	Amy	
3		345	Craig	
4		987	Helen	
5		987	Helen	
6		876	Irene	
7		543	Craig	

	R 2	SID	R 2	SNAME
1		123	Amy	
2		345	Craig	
3		987	Helen	
4		876	Irene	
5		543	Craig	

	R2	SID	R2	SNAME
1		123	Amy	
2		123	Amy	
3		345	Craig	
4		987	Helen	
5		987	Helen	
6		876	Irene	
7		543	Craig	

	R2	SID	R2	SNAME
1		123	Amy	
2		345	Craig	
3		987	Helen	
4		876	Irene	
5		543	Craig	

	R2	SID	R2	CNAME	R2	MAJOR	R2	DECISION
1		123	Stanford	CS			Y	
2		123	Stanford	EE			N	
3		123	Berkeley	CS			Y	
4		123	Cornell	EE			Y	
5		234	Berkeley	biology			N	
6		345	MIT	bioengineering			Y	
7		345	Cornell	bioengineering			N	
8		345	Cornell	CS			Y	
9		345	Cornell	EE			N	
10		678	Stanford	history			Y	
11		987	Stanford	CS			Y	
12		987	Berkeley	CS			Y	
13		876	Stanford	CS			N	
14		876	MIT	biology			Y	
15		876	MIT	marine biology			N	
16		876	Stanford	history			Y	
17		765	Cornell	history			N	
18		765	Cornell	psychology			Y	
19		765	MIT	CS			N	

SQL

```
SELECT      Student.sID, sName
FROM        Student, Apply
WHERE       Student.sID =
            Apply.sID
            AND
            major = 'CS';
```

Result

	R2	SID	R2	SNAME
1		123	Amy	
2		123	Amy	
3		345	Craig	
4		987	Helen	
5		987	Helen	
6		876	Irene	
7		543	Craig	

SQL

```
SELECT DISTINCT Student.sID,  
                sName  
FROM      Student, Apply  
WHERE     Student.sID =  
          Apply.sID  
AND       major = 'CS';
```

Result

	R Z	SID	R Z	SNAME
1		123		Amy
2		987		Helen
3		345		Craig
4		543		Craig
5		876		Irene

SQL

```
SELECT      sName
FROM        Student
WHERE       sID IN
(
    SELECT   sID
    FROM      Apply
    WHERE     major = 'CS'
);
```

Result

	RZ	SNAME
1		Amy
2		Craig
3		Helen
4		Irene
5		Craig

SQL

```
SELECT    sName  
FROM      Student, Apply  
WHERE     Student.sID =  
          Apply.sID  
AND       major = 'CS';
```

SQL

```
SELECT      sName
FROM        Student, Apply
WHERE       Student.sID =
            Apply.sID
            AND
            major = 'CS';
```

Result

	RZ	SNAME
1		Amy
2		Amy
3		Craig
4		Helen
5		Helen
6		Irene
7		Craig

SQL

```
SELECT DISTINCT sName
FROM      Student, Apply
WHERE     Student.sID =
          Apply.sID
AND       major = 'CS';
```

Result

	R/R	SNAME
1		Craig
2		Helen
3		Amy
4		Irene

SQL

```
SELECT DISTINCT sName
FROM      Student, Apply
WHERE     Student.sID =
          Apply.sID
AND       major = 'CS';
```

```
SELECT DISTINCT Student.sID,
                 sName
FROM      Student, Apply
WHERE     Student.sID =
          Apply.sID
AND       major = 'CS';
```

Result

	 SNAME
1	Craig
2	Helen
3	Amy
4	Irene

	 SID	 SNAME
1	123	Amy
2	987	Helen
3	345	Craig
4	543	Craig
5	876	Irene

SQL

```
SELECT      GPA
FROM        Student
WHERE       sID IN
(
  SELECT     sID
  FROM        Apply
  WHERE       major = 'CS'
);
```

Result

	R/R	GPA
1		3.9
2		3.5
3		3.7
4		3.9
5		3.4

SQL

```
SELECT GPA
FROM Student, Apply
WHERE Student.sID =
       Apply.sID
AND   major = 'CS';
```

Result

	R/R	GPA
1		3.9
2		3.9
3		3.5
4		3.7
5		3.7
6		3.9
7		3.4

SQL

```
SELECT DISTINCT GPA
FROM      Student, Apply
WHERE     Student.sID =
          Apply.sID
AND       major = 'CS';
```

Result

	RZ	GPA
1		3.7
2		3.5
3		3.9
4		3.4

SQL

```
SELECT DISTINCT GPA
FROM      Student, Apply
WHERE     Student.sID =
          Apply.sID
AND       major = 'CS';
```

```
SELECT      GPA
FROM        Student
WHERE       sID IN
(
  SELECT     sID
  FROM       Apply
  WHERE      major = 'CS'
);
```

Result

	RZ	GPA
1		3.7
2		3.5
3		3.9
4		3.4

	RZ	GPA
1		3.9
2		3.5
3		3.7
4		3.9
5		3.4

SQL

```
SELECT      sID, sName
FROM        Student
WHERE       sID IN
(
    SELECT      sID
    FROM        Apply
    WHERE       major = 'CS'
)
AND         sID NOT IN
(
    SELECT      sID
    FROM        Apply
    WHERE       major = 'EE'
);
```

SQL

```
SELECT      sID, sName
FROM        Student
WHERE       sID IN
(
    SELECT      sID
    FROM        Apply
    WHERE       major = 'CS'
)
AND         sID NOT IN
(
    SELECT      sID
    FROM        Apply
    WHERE       major = 'EE'
);
```

Result

	R 2	SID	R 2	SNAME
1		543		Craig
2		987		Helen
3		876		Irene

SQL

```
SELECT      sID, sName
FROM        Student
WHERE       sID IN
(
    SELECT      sID
    FROM        Apply
    WHERE       major = 'CS'
)
AND NOT sID IN
(
    SELECT      sID
    FROM        Apply
    WHERE       major = 'EE'
);
```

Result

	R 2	SID	R 2	SNAME
1		543		Craig
2		987		Helen
3		876		Irene

SQL

```
SELECT      cName, state
FROM        College C1
WHERE       EXISTS
(
  SELECT      *
  FROM        College C2
  WHERE       C2.state = C1.state
);
```

SQL

```
SELECT      cName, state
FROM        College C1
WHERE       EXISTS
(
  SELECT      *
  FROM        College C2
  WHERE       C2.state = C1.state
);
```

Result

	RZ CNAME	RZ STATE
1	Berkeley	CA
2	Stanford	CA
3	MIT	MA
4	Cornell	NY

SQL

```
SELECT      cName, state
FROM        College C1
WHERE       EXISTS
(
  SELECT      *
  FROM        College C2
  WHERE       C2.state = C1.state
  AND        C1.cName <> C2.cName
);
```

SQL

```
SELECT      cName, state
FROM        College C1
WHERE       EXISTS
(
  SELECT      *
  FROM        College C2
  WHERE       C2.state = C1.state
  AND        C1.cName <> C2.cName
);
```

Result

	RZ CNAME	RZ STATE
1	Berkeley	CA
2	Stanford	CA

SQL

```
SELECT      cName
FROM        College c1
WHERE       NOT EXISTS
(
  SELECT      *
  FROM        College c2
  WHERE       c2.enrollment >
              c1.enrollment
);
```

SQL

```
SELECT      cName
FROM        College c1
WHERE       NOT EXISTS
(
  SELECT      *
  FROM        College c2
  WHERE       c2.enrollment >
              c1.enrollment
);
```

Result

	CNAME
1	Berkeley

SQL

```
SELECT      sName
FROM        Student s1
WHERE       NOT EXISTS
(
  SELECT      *
  FROM        Student s2
  WHERE       S2.GPA > S1.GPA
);
```

SQL

```
SELECT      sName
FROM        Student s1
WHERE       NOT EXISTS
(
    SELECT    *
    FROM      Student s2
    WHERE     S2.GPA > S1.GPA
);
```

Result

	R2	SNAME
1		Irene
2		Amy
3		Amy
4		Doris

SQL

```
SELECT      sName, GPA
FROM        Student s1
WHERE       NOT EXISTS
(
  SELECT      *
  FROM        Student s2
  WHERE       S2.GPA > S1.GPA
);
```

Result

	R1 Z	SNAME	R2 Z	GPA
1		Irene		3.9
2		Amy		3.9
3		Amy		3.9
4		Doris		3.9

SQL

```
SELECT    S1.sName, S1.GPA  
FROM      Student S1, Student S2  
WHERE     S1.GPA > S2.GPA;
```


SQL

```
SELECT      S1.sName, S1.GPA
FROM        Student S1, Student S2
WHERE       S1.GPA > S2.GPA;
```

Result

	R1	SNAME	R2	GPA
1	Amy			3.9
2	Amy			3.9
3	Amy			3.9
4	Amy			3.9
5	Amy			3.9
6	Amy			3.9
7	Amy			3.9
8	Amy			3.9
9	Amy			3.9
10	Amy			3.9
11	Amy			3.9
12	Amy			3.9
13	Amy			3.9
14	Amy			3.9
15	Amy			3.9
16	Amy			3.9
				3.9
				3.9
				3.9

SQL

```
SELECT DISTINCT S1.sName, S1.GPA
FROM      Student S1, Student S2
WHERE      S1.GPA > S2.GPA;
```

Result

	R2	SNAME	R2	GPA
1		Amy		3.9
2		Irene		3.9
3		Fay		3.8
4		Helen		3.7
5		Gary		3.4
6		Bob		3.6
7		Craig		3.5
8		Craig		3.4
9		Doris		3.9

SQL

```
SELECT      sName, GPA
FROM        Student
WHERE       GPA >= ALL
(
    SELECT      GPA
    FROM        Student
);
```

SQL

```
SELECT      sName, GPA
FROM        Student
WHERE       GPA >= ALL
(
    SELECT      GPA
    FROM        Student
);
```

Result

	R2	SNAME	R2	GPA
1		Irene		3.9
2		Amy		3.9
3		Amy		3.9
4		Doris		3.9

SQL

```
SELECT      sName
FROM        Student s1
WHERE       GPA > ALL
(
  SELECT      GPA
  FROM        Student s2
  WHERE       S2.sID <> S1.sID
);
```

SQL

```
SELECT      sName
FROM        Student s1
WHERE       GPA > ALL
(
  SELECT      GPA
  FROM        Student s2
  WHERE       S2.sID <> S1.sID
);
```

Result

SNAME

SQL

```
SELECT      cName
FROM        College C1
WHERE       enrollment > ALL
(
  SELECT     enrollment
  FROM       College C2
  WHERE      C2.cName <> C1.cName
);
```

Result

	CNAME
1	Berkeley

SQL

```
SELECT      cName
FROM        College c1
WHERE NOT enrollment <= ANY
(
  SELECT      enrollment
  FROM        College c2
  WHERE      C2.cName <> C1.cName
);
```


SQL

```
SELECT      cName
FROM        College c1
WHERE NOT enrollment <= ANY
(
  SELECT      enrollment
  FROM        College c2
  WHERE      C2.cName <> C1.cName
);
```

Result

	R 2	CNAME
	1	Berkeley

SQL

```
SELECT      sID, sName, sizeHS  
FROM        Student  
WHERE       sizeHS > ANY  
(  
    SELECT   sizeHS  
    FROM     Student  
);
```

SQL

```
SELECT      sID, sName, sizeHS
FROM        Student
WHERE       sizeHS > ANY
(
    SELECT   sizeHS
    FROM     Student
);
```

Result

	ASC	SID	ASC	SNAME	ASC	SIZEHS
1		543		Craig		2000
2		567		Edward		2000
3		234		Bob		1500
4		765		Jay		1500
5		456		Doris		1000
6		654		Amy		1000
7		123		Amy		1000
8		987		Helen		800
9		789		Gary		800
10		345		Craig		500
11		876		Irene		400

SQL

```
SELECT      sID, sName, sizeHS  
FROM        Student s1  
WHERE       EXISTS  
(  
    SELECT      *  
    FROM        Student s2  
    WHERE       s2.sizeHS <  
                s1.sizeHS  
);
```

SQL

```
SELECT      sID, sName, sizeHS
FROM        Student s1
WHERE       EXISTS
(
    SELECT      *
    FROM        Student s2
    WHERE       s2.sizeHS <
                s1.sizeHS
);
```

Result

	PK SID	PK SNAME	PK SIZEHS
1	543	Craig	2000
2	567	Edward	2000
3	234	Bob	1500
4	765	Jay	1500
5	456	Doris	1000
6	654	Amy	1000
7	123	Amy	1000
8	987	Helen	800
9	789	Gary	800
10	345	Craig	500
11	876	Irene	400

SQL

```
SELECT      sID, sName
FROM        Student
WHERE       sID = ANY
(
    SELECT      sID
    FROM        Apply
    WHERE       major = 'CS'
)
AND         sID <> ANY
(
    SELECT      sID
    FROM        Apply
    WHERE       major = 'EE'
);
```

SQL

```
SELECT      sID, sName
FROM        Student
WHERE       sID IN
(
    SELECT      sID
    FROM        Apply
    WHERE       major = 'CS'
)
AND NOT sID IN
(
    SELECT      sID
    FROM        Apply
    WHERE       major = 'EE'
);
```

Result

	R Z	SID	R Z	SNAME
1		543		Craig
2		987		Helen
3		876		Irene

SQL

```
SELECT      sID, sName
FROM        Student
WHERE       sID = ANY
(
    SELECT      sID
    FROM        Apply
    WHERE       major = 'CS'
)
AND         sID <> ANY
(
    SELECT      sID
    FROM        Apply
    WHERE       major = 'EE'
);
```


SQL

```
SELECT      sID, sName
FROM        Student
WHERE       sID = ANY
(
    SELECT      sID
    FROM        Apply
    WHERE       major = 'CS'
)
AND         sID <> ANY
(
    SELECT      sID
    FROM        Apply
    WHERE       major = 'EE'
);
```

Result

	R/R	SID	R/R	SNAME
1		123		Amy
2		345		Craig
3		987		Helen
4		876		Irene
5		543		Craig

SQL

```
SELECT      sID, sName
FROM        Student
WHERE       sID = ANY
(
    SELECT      sID
    FROM        Apply
    WHERE       major = 'CS'
)
AND NOT     sID = ANY
(
    SELECT      sID
    FROM        Apply
    WHERE       major = 'EE'
);
```

SQL

```
SELECT      sID, sName
FROM        Student
WHERE       sID = ANY
(
    SELECT      sID
    FROM        Apply
    WHERE       major = 'CS'
)
AND NOT     sID = ANY
(
    SELECT      sID
    FROM        Apply
    WHERE       major = 'EE'
);
```

Result

	R/R	SID	R/R	SNAME
1		543		Craig
2		987		Helen
3		876		Irene

Subqueries in FROM and SELECT

SELECT
FROM
WHERE

$A_1, A_2, \dots, A_n$
 $R_1, R_2, \dots, R_m$
condition

Expressions involving
“subqueries”: nested
Select statements

Demo: simple college admissions database

College(cName, state, enrollment)

Student(sID, sName, GPA, sizeHS)

Apply(sID, cName, major, decision)

SQL

```
SELECT      SID, sName, GPA,  
            GPA*(sizeHS/1000.0)  
            AS scaledGPA  
FROM        Student  
WHERE       GPA*(sizeHS/1000.0)  
            - GPA > 1.0  
OR          GPA -  
            GPA*(sizeHS/1000.0) > 1.0;
```

SQL

```
SELECT      SID, sName, GPA,  
            GPA*(sizeHS/1000.0)  
            AS scaledGPA  
FROM        Student  
WHERE       GPA*(sizeHS/1000.0)  
            - GPA > 1.0  
OR          GPA -  
            GPA*(sizeHS/1000.0) > 1.0;
```

Result

	R2	SID	R2	SNAME	R2	GPA	R2	SCALEDGPA
1		234		Bob		3.6		5.4
2		345		Craig		3.5		1.75
3		567		Edward		2.9		5.8
4		678		Fay		3.8		0.76
5		876		Irene		3.9		1.56
6		765		Jay		2.9		4.35
7		543		Craig		3.4		6.8

SQL

```
SELECT      SID, sName, GPA,
            GPA*(sizeHS/1000.0)
            AS scaledGPA
FROM        Student
WHERE       ABS(GPA*(sizeHS/1000.0)
              - GPA) > 1.0;
```

Result

	R2	SID	R2	SNAME	R2	GPA	R2	SCALEDGPA
1		234		Bob		3.6		5.4
2		345		Craig		3.5		1.75
3		567		Edward		2.9		5.8
4		678		Fay		3.8		0.76
5		876		Irene		3.9		1.56
6		765		Jay		2.9		4.35
7		543		Craig		3.4		6.8

SQL

```
SELECT      *
FROM (
    SELECT  sID, sName, GPA,
            GPA*(sizeHS/1000.0)
            AS scaledGPA
    FROM    Student
) G
WHERE ABS (G.scaledGPA - GPA)
        > 1.0;
```

SQL

```
SELECT      *
FROM (
    SELECT  SID, sName, GPA,
            GPA*(sizeHS/1000.0)
            AS scaledGPA
    FROM    Student
) G
WHERE ABS (G.scaledGPA - GPA)
        > 1.0;
```

Result

	R2	SID	R2	SNAME	R2	GPA	R2	SCALEDGPA
1		234		Bob		3.6		5.4
2		345		Craig		3.5		1.75
3		567		Edward		2.9		5.8
4		678		Fay		3.8		0.76
5		876		Irene		3.9		1.56
6		765		Jay		2.9		4.35
7		543		Craig		3.4		6.8

SQL

```
SELECT    College.cName,  
          state, GPA  
FROM      College,  
          Apply, Student  
WHERE     College.cName =  
          Apply.cName  
          AND Apply.sID =  
          Student.sID  
          AND GPA >= ALL (  
      SELECT GPA  
      FROM    Student, Apply  
      WHERE   Student.sID =  
              Apply.sID  
              AND Apply.cName =  
              College.cName  
);
```

SQL

```
SELECT      College.cName,  
            state, GPA  
FROM        College,  
            Apply, Student  
WHERE       College.cName =  
            Apply.cName  
            AND Apply.sID =  
            Student.sID  
            AND GPA >= ALL (  
            SELECT      GPA  
            FROM        Student, Apply  
            WHERE       Student.sID =  
            Apply.sID  
            AND Apply.cName =  
            College.cName  
            );
```

Result

	 CNAME	 STATE	 GPA
1	Cornell	NY	3.9
2	Berkeley	CA	3.9
3	Stanford	CA	3.9
4	Stanford	CA	3.9
5	MIT	MA	3.9
6	MIT	MA	3.9
7	Stanford	CA	3.9

SQL

```
SELECT DISTINCT College.cName,  
                state, GPA  
FROM            College,  
                Apply, Student  
WHERE           College.cName =  
                Apply.cName  
AND             Apply.sID =  
                Student.sID  
AND            GPA >= ALL (  
    SELECT      GPA  
    FROM        Student, Apply  
    WHERE       Student.sID =  
                Apply.sID  
    AND         Apply.cName =  
                College.cName  
);
```

Result

	 CNAME	 STATE	 GPA
1	Cornell	NY	3.9
2	MIT	MA	3.9
3	Berkeley	CA	3.9
4	Stanford	CA	3.9

SQL

```
SELECT      cName, state,
( SELECT DISTINCT GPA
  FROM      Apply, Student
 WHERE      College.cName =
            Apply.cName
            AND Apply.sID =
            Student.sID
            AND GPA >= ALL (
              SELECT GPA
                FROM   Student, Apply
              WHERE    Student.sID =
                    Apply.sID
                    AND Apply.cName =
                    College.cName )
) AS GPA
FROM        College;
```

SQL

```
SELECT      cName, state,
( SELECT DISTINCT GPA
  FROM      Apply, Student
  WHERE      College.cName =
              Apply.cName
            AND Apply.sID =
              Student.sID
            AND GPA >= ALL (
              SELECT GPA
                FROM   Student, Apply
                WHERE   Student.sID =
                      Apply.sID
                    AND Apply.cName =
                      College.cName )
) AS GPA
FROM        College;
```

Result

	 CNAME	 STATE	 GPA
1	Stanford	CA	3.9
2	Berkeley	CA	3.9
3	MIT	MA	3.9
4	Cornell	NY	3.9

SQL

```
SELECT      cName, state
( SELECT DISTINCT sName
  FROM      Apply, Student
  WHERE     College.cName =
            Apply.cName
            AND Apply.sID =
            Student.sID )
  AS sName
FROM        College;
```


SQL

```
SELECT      cName, state
( SELECT DISTINCT sName
  FROM      Apply, Student
  WHERE     College.cName =
            Apply.cName
            AND Apply.sID =
            Student.sID )
  AS sName
FROM        College;
```

Result

```
ORA-00936: 누락된 표현식
00936. 00000 - "missing expression"
*Cause:
*Action:
530행, 3열에서 오류 발생
```

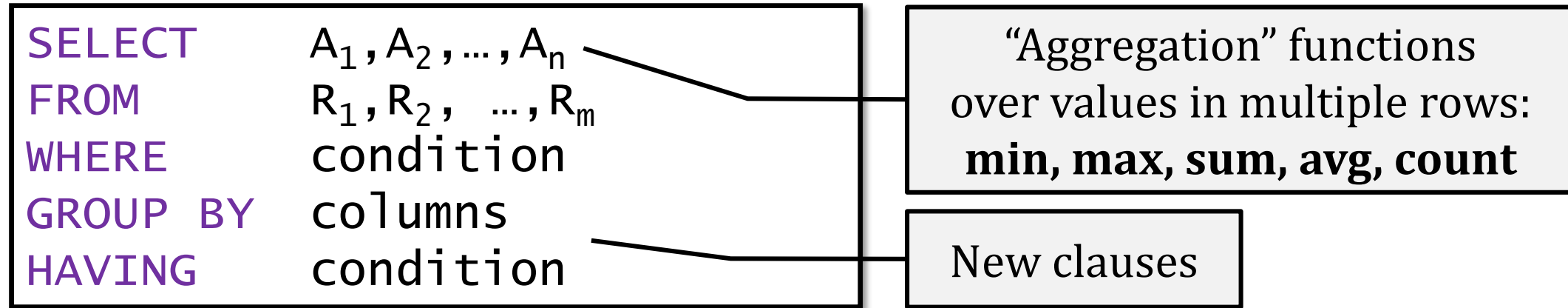

Aggregation

SELECT $A_1, A_2, \dots, A_n$
FROM $R_1, R_2, \dots, R_m$
WHERE condition



“Aggregation” functions
over values in multiple rows:
min, max, sum, avg, count

Aggregation



Demo: simple college admissions database

College(cName, state, enrollment)

Student(sID, sName, GPA, sizeHS)

Apply(sID, cName, major, decision)

SQL

```
SELECT *  
FROM Student;
```

Result

	A2	SID	A2	SNAME	A2	GPA	A2	SIZES
1		123		Amy		3.9		1000
2		234		Bob		3.6		1500
3		345		Craig		3.5		500
4		456		Doris		3.9		1000
5		567		Edward		2.9		2000
6		678		Fay		3.8		200
7		789		Gary		3.4		800
8		987		Helen		3.7		800
9		876		Irene		3.9		400
10		765		Jay		2.9		1500
11		654		Amy		3.9		1000
12		543		Craig		3.4		2000

SQL

```
SELECT      AVG(GPA)
FROM        Student;
```

Result

	Avg(GPA)
1	3.5666666666666666666666666666666667

SQL

```
SELECT *
FROM Student, Apply
WHERE Student.sID =
      Apply.sID
AND major = 'CS';
```

Result

	SID	SNAME	GPA	SIZEHS	SID_1	CNAME	MAJOR	DECISION
1	123	Amy	3.9	1000	123	Berkeley	CS	Y
2	123	Amy	3.9	1000	123	Stanford	CS	Y
3	345	Craig	3.5	500	345	Cornell	CS	Y
4	987	Helen	3.7	800	987	Berkeley	CS	Y
5	987	Helen	3.7	800	987	Stanford	CS	Y
6	876	Irene	3.9	400	876	Stanford	CS	N
7	543	Craig	3.4	2000	543	MIT	CS	N

SQL

```
SELECT MIN(GPA)
FROM Student, Apply
WHERE Student.sID =
      Apply.sID
AND major = 'CS';
```

Result

	MIN(GPA)
1	3.4

SQL

```
SELECT *
FROM Student, Apply
WHERE Student.sID =
        Apply.sID
AND major = 'CS';
```

Result

	SID	SNAME	GPA	SIZEHS	SID_1	CNAME	MAJOR	DECISION
1	123	Amy	3.9	1000	123	Berkeley	CS	Y
2	123	Amy	3.9	1000	123	Stanford	CS	Y
3	345	Craig	3.5	500	345	Cornell	CS	Y
4	987	Helen	3.7	800	987	Berkeley	CS	Y
5	987	Helen	3.7	800	987	Stanford	CS	Y
6	876	Irene	3.9	400	876	Stanford	CS	N
7	543	Craig	3.4	2000	543	MIT	CS	N

SQL

```
SELECT AVG(GPA)
FROM Student, Apply
WHERE Student.sID =
        Apply.sID
AND major = 'CS';
```

Result

	AVG(GPA)
1	3.71428571428571428571428571428571428571

SQL

```
SELECT *
FROM Student
WHERE SID IN
( SELECT SID
  FROM Apply
  WHERE major = 'CS'
);
```

Result

	AZ	SID	AZ	SNAME	AZ	GPA	AZ	SIZEHS
1		123		Amy		3.9		1000
2		345		Craig		3.5		500
3		987		Helen		3.7		800
4		876		Irene		3.9		400
5		543		Craig		3.4		2000

SQL

```
SELECT AVG(GPA)
FROM Student
WHERE SID IN
( SELECT SID
  FROM Apply
  WHERE major = 'CS'
);
```

Result

	AZ	AVG(GPA)
1		3.68

SQL

```
SELECT *  
FROM College  
WHERE enrollment > 15000;
```

Result

	CNAME	STATE	ENROLLMENT
1	Berkeley	CA	36000
2	Cornell	NY	21000

SQL

```
SELECT COUNT(*)  
FROM College  
WHERE enrollment > 15000;
```

Result

	COUNT(*)
1	2

SQL

```
SELECT      COUNT(*)
FROM        Apply
WHERE       cName = 'Cornell';
```

Result

	1	COUNT(*)
	1	6

SQL

```
SELECT      COUNT(DISTINCT sID)
FROM        Apply
WHERE       cName = 'Cornell';
```

Result

	1	COUNT(DISTINCTSID)
	1	3

SQL

```
SELECT      SID
FROM        Apply
WHERE       cName = 'Cornell';
```

Result

	RZ	SID
1		123
2		345
3		345
4		345
5		765
6		765

SQL

```
SELECT      *
FROM        Student s1
WHERE
( SELECT COUNT(*)
  FROM      Student s2
  WHERE     S2.sID <> S1.sID
    AND     S2.GPA = S1.GPA
) =
( SELECT COUNT(*)
  FROM      Student s2
  WHERE     S2.sID <> S1.sID
    AND     S2.sizeHS = S1.sizeHS
);
```

Result

	PK SID	PK SNAME	PK GPA	PK SIZEHS
1	345	Craig	3.5	500
2	567	Edward	2.9	2000
3	678	Fay	3.8	200
4	789	Gary	3.4	800
5	765	Jay	2.9	1500
6	543	Craig	3.4	2000

SQL

```
SELECT  CS.avgGPA - NonCS.avgGPA
FROM
(SELECT  AVG(GPA) AS avgGPA
FROM      Student
WHERE     SID IN
        (SELECT  SID
FROM        Apply
WHERE       major = 'CS')) CS,
(SELECT  AVG(GPA) AS avgGPA
FROM      Student
WHERE     SID NOT IN
        (SELECT  SID
FROM        Apply
WHERE       major = 'CS')) nonCS;
```

SQL

```
SELECT  CS.avgGPA - NonCS.avgGPA
FROM
(SELECT  AVG(GPA) AS avgGPA
FROM      Student
WHERE     SID IN
        (SELECT  SID
FROM        Apply
WHERE       major = 'CS')) CS,
(SELECT  AVG(GPA) AS avgGPA
FROM      Student
WHERE     SID NOT IN
        (SELECT  SID
FROM        Apply
WHERE       major = 'CS')) nonCS;
```

Result

	CS.AVG GPA - NONCS.AVG GPA
1	0.19428571428571428571428571428571428571

SQL

```
SELECT
( SELECT  AVG(GPA) AS avgGPA
  FROM    Student
  WHERE   SID IN
        ( SELECT  SID
          FROM    Apply
          WHERE   major = 'CS' ) ) -
( SELECT  AVG(GPA) AS avgGPA
  FROM    Student
  WHERE   SID NOT IN
        ( SELECT  SID
          FROM    Apply
          WHERE   major = 'CS' ) ) D
FROM    Student;
```

Result

	RZ	D
1		0.19428571428571428571428571428571428571
2		0.19428571428571428571428571428571428571
3		0.19428571428571428571428571428571428571
4		0.19428571428571428571428571428571428571
5		0.19428571428571428571428571428571428571
6		0.19428571428571428571428571428571428571
7		0.19428571428571428571428571428571428571
8		0.19428571428571428571428571428571428571
9		0.19428571428571428571428571428571428571
10		0.19428571428571428571428571428571428571
11		0.19428571428571428571428571428571428571
12		0.19428571428571428571428571428571428571

SQL

```
SELECT DISTINCT
( SELECT  AVG(GPA) AS avgGPA
  FROM    Student
  WHERE   SID IN
        ( SELECT  SID
          FROM    Apply
          WHERE   major = 'CS' ) ) -
( SELECT  AVG(GPA) AS avgGPA
  FROM    Student
  WHERE   SID NOT IN
        ( SELECT  SID
          FROM    Apply
          WHERE   major = 'CS' ) ) D
FROM Student;
```

Result

	R	D
1	0.19428571428571428571428571428571428571	

SQL

```
SELECT      cName, COUNT(*)  
FROM        Apply  
GROUP BY    cName;
```

Result

	 CNAME	 COUNT(*)
1	Stanford	6
2	Cornell	6
3	Berkeley	3
4	MIT	4

SQL

```
SELECT *  
FROM Apply  
ORDER BY cName;
```

Result

	Rank	SID	Rank	CNAME	Rank	MAJOR	Rank	DECISION
1		123		Berkeley		CS		Y
2		987		Berkeley		CS		Y
3		234		Berkeley		biology		N
4		765		Cornell		history		N
5		765		Cornell		psychology		Y
6		345		Cornell		bioengineering		N
7		345		Cornell		EE		N
8		345		Cornell		CS		Y
9		123		Cornell		EE		Y
10		876		MIT		marine biology		N
11		876		MIT		biology		Y
12		543		MIT		CS		N
13		345		MIT		bioengineering		Y
14		765		Stanford		history		Y
15		876		Stanford		CS		N
16		987		Stanford		CS		Y
17		678		Stanford		history		Y
18		123		Stanford		CS		Y
19		123		Stanford		EE		N

SQL

```
SELECT      cName, COUNT(*)  
FROM        Apply  
GROUP BY    cName;
```

Result

	 CNAME	 COUNT(*)
1	Stanford	6
2	Cornell	6
3	Berkeley	3
4	MIT	4

SQL

```
SELECT      state, SUM(enrollment)
FROM        College
GROUP BY    state;
```

Result

	STATE	SUM(ENROLLMENT)
1	CA	51000
2	MA	10000
3	NY	21000

SQL

```
SELECT      cName, major,  
            MIN(GPA), MAX(GPA)  
FROM        Student, Apply  
WHERE       Student.sID =  
            Apply.sID  
GROUP BY    cName, major;
```

SQL

```
SELECT      cName, major, GPA
FROM        Student, Apply
WHERE       Student.sID =
            Apply.sID
ORDER BY    cName, major;
```

Result

	 CNAME	 MAJOR	 GPA
1	Berkeley	CS	3.9
2	Berkeley	CS	3.7
3	Berkeley	biology	3.6
4	Cornell	CS	3.5
5	Cornell	EE	3.9
6	Cornell	EE	3.5
7	Cornell	bioengineering	3.5
8	Cornell	history	2.9
9	Cornell	psychology	2.9
10	MIT	CS	3.4
11	MIT	bioengineering	3.5
12	MIT	biology	3.9
13	MIT	marine biology	3.9
14	Stanford	CS	3.9
15	Stanford	CS	3.7
16	Stanford	CS	3.9
17	Stanford	EE	3.9
18	Stanford	history	2.9
19	Stanford	history	3.8

SQL

```
SELECT      cName, major,
            MIN(GPA), MAX(GPA)
FROM        Student, Apply
WHERE       Student.sID =
            Apply.sID
GROUP BY    cName, major;
```

Result

	 CNAME	 MAJOR	 MIN(GPA)	 MAX(GPA)
1	Berkeley	CS	3.7	3.9
2	Cornell	psychology	2.9	2.9
3	MIT	CS	3.4	3.4
4	MIT	marine biology	3.9	3.9
5	Cornell	EE	3.5	3.9
6	Stanford	CS	3.7	3.9
7	Stanford	history	2.9	3.8
8	Stanford	EE	3.9	3.9
9	Cornell	CS	3.5	3.5
10	Berkeley	biology	3.6	3.6
11	Cornell	bioengineering	3.5	3.5
12	MIT	bioengineering	3.5	3.5
13	MIT	biology	3.9	3.9
14	Cornell	history	2.9	2.9

SQL

```
SELECT      MAX(mx-mn)
FROM
(
  SELECT      cName, major,
              MIN(GPA) AS mn,
              MAX(GPA) AS mx
  FROM        Student, Apply
  WHERE       Student.sID =
              Apply.sID
  GROUP BY   cName, major
) M;
```

Result

	MAX(MX-MN)
1	0.9

SQL

```
SELECT      Student.sID,  
            COUNT(DISTINCT cName)  
FROM        Student, Apply  
WHERE       Student.sID =  
            Apply.sID  
GROUP BY    Student.sID;
```

SQL

```
SELECT      Student.sID, cName
FROM        Student, Apply
WHERE       Student.sID =
           Apply.sID
ORDER BY    Student.sID;
```

Result

	R 2	SID	R 2	CNAME
1		123		Stanford
2		123		Stanford
3		123		Berkeley
4		123		Cornell
5		234		Berkeley
6		345		MIT
7		345		Cornell
8		345		Cornell
9		345		Cornell
10		543		MIT
11		678		Stanford
12		765		Stanford
13		765		Cornell
14		765		Cornell
15		876		Stanford
16		876		MIT
17		876		MIT
18		987		Berkeley
19		987		Stanford

SQL

```
SELECT      Student.sID,  
            COUNT(DISTINCT cName)  
FROM        Student, Apply  
WHERE       Student.sID =  
            Apply.sID  
GROUP BY    Student.sID;
```

Result

	RANK	SID	RANK	COUNT(DISTINCTCNAME)
1		123		3
2		678		1
3		543		1
4		987		2
5		876		2
6		765		2
7		234		1
8		345		2

SQL

```
SELECT      Student.sID,  
            COUNT(DISTINCT cName)  
FROM        Student, Apply  
WHERE       Student.sID =  
            Apply.sID  
GROUP BY    Student.sID  
UNION  
SELECT      sID, 0  
FROM        Student  
WHERE       sID NOT IN  
( SELECT sID FROM Apply );
```

SQL

```
SELECT      Student.sID,  
            COUNT(DISTINCT cName)  
FROM        Student, Apply  
WHERE       Student.sID =  
            Apply.sID  
GROUP BY    Student.sID  
UNION  
SELECT      sID, 0  
FROM        Student  
WHERE       sID NOT IN  
( SELECT sID FROM Apply );
```

Result

	R Z	SID	R Z	COUNT(DISTINCTCNAME)
1		123		3
2		234		1
3		345		2
4		456		0
5		543		1
6		567		0
7		654		0
8		678		1
9		765		2
10		789		0
11		876		2
12		987		2

SQL

```
SELECT      cName  
FROM        Apply  
GROUP BY    cName  
HAVING      COUNT(*) < 5;
```

SQL

```
SELECT      cName  
FROM        Apply  
GROUP BY    cName  
HAVING      COUNT(*) < 5;
```

Result

	RZ	CNAME
1		Berkeley
2		MIT

SQL

```
SELECT      cName
FROM        Apply A1
WHERE       5 >
(
  SELECT COUNT(*)
  FROM Apply A2
  WHERE A2.cName = A1.cName
);
```

Result

	R2	CNAME
1		Berkeley
2		Berkeley
3		MIT
4		Berkeley
5		MIT
6		MIT
7		MIT

SQL

```
SELECT      DISTINCT cName
FROM        Apply A1
WHERE       5 >
(
  SELECT COUNT(*)
  FROM Apply A2
  WHERE A2.cName = A1.cName
);
```

Result

	RZ	CNAME
1		Berkeley
2		MIT

SQL

```
SELECT      cName
FROM        Apply
GROUP BY    cName
HAVING      COUNT(*) < 5;
```

Result

	RZ	CNAME
1		Berkeley
2		MIT

SQL

```
SELECT      cName
FROM        Apply
GROUP BY    cName
HAVING      COUNT(DISTINCT sID) < 5;
```

Result

	RZ	CNAME
1		Cornell
2		MIT
3		Berkeley

SQL

```
SELECT      major
FROM        Student, Apply
WHERE       Student.sID =
            Apply.sID
GROUP BY    major
HAVING      MAX(GPA) <
( SELECT AVG(GPA) FROM Student );
```

SQL

```
SELECT      major
FROM        Student, Apply
WHERE       Student.sID =
           Apply.sID
GROUP BY    major
HAVING      MAX(GPA) <
( SELECT AVG(GPA) FROM Student );
```

Result

	MAJOR
1	bioengineering
2	psychology

NULL Values

SELECT	$A_1, A_2, \dots, A_n$
FROM	$R_1, R_2, \dots, R_m$
WHERE	condition

Demo: simple college admissions database

College(cName, state, enrollment)

Student(sID, sName, GPA, sizeHS)

Apply(sID, cName, major, decision)

NULL values:
undefined or unknown

SQL

```
SELECT *  
FROM Student;
```

Result

	SID	SNAME	GPA	SIZEHS
1	123	Amy	3.9	1000
2	234	Bob	3.6	1500
3	345	Craig	3.5	500
4	456	Doris	3.9	1000
5	567	Edward	2.9	2000
6	678	Fay	3.8	200
7	789	Gary	3.4	800
8	987	Helen	3.7	800
9	876	Irene	3.9	400
10	765	Jay	2.9	1500
11	654	Amy	3.9	1000
12	543	Craig	3.4	2000

SQL

```
INSERT INTO Student VALUES (432,  
'Kevin', null, 1500);  
INSERT INTO Student VALUES (321,  
'Lori', null, 2500);
```

Result

	PK	SID	PK	SNAME	PK	GPA	PK	SIZEHS
1		123		Amy		3.9		1000
2		234		Bob		3.6		1500
3		345		Craig		3.5		500
4		456		Doris		3.9		1000
5		567		Edward		2.9		2000
6		678		Fay		3.8		200
7		789		Gary		3.4		800
8		987		Helen		3.7		800
9		876		Irene		3.9		400
10		765		Jay		2.9		1500
11		654		Amy		3.9		1000
12		543		Craig		3.4		2000
13		432		Kevin		(null)		1500
14		321		Lori		(null)		2500

SQL

```
SELECT      SID, sName, GPA
FROM        Student
WHERE       GPA > 3.5;
```

Result

	RZ	SID	RZ	SNAME	RZ	GPA
1		123		Amy		3.9
2		234		Bob		3.6
3		456		Doris		3.9
4		678		Fay		3.8
5		987		Helen		3.7
6		876		Irene		3.9
7		654		Amy		3.9

SQL

```
SELECT    sID, sName, GPA
FROM      Student
WHERE     GPA <= 3.5;
```

Result

	R Z	SID	R Z	SNAME	R Z	GPA
1		345		Craig		3.5
2		567		Edward		2.9
3		789		Gary		3.4
4		765		Jay		2.9
5		543		Craig		3.4

SQL

```
SELECT    SID, sName, GPA
FROM      Student
WHERE     GPA > 3.5 OR
          GPA <= 3.5;
```

Result

	PK	SID	PK	SNAME	PK	GPA
1		123		Amy		3.9
2		234		Bob		3.6
3		345		Craig		3.5
4		456		Doris		3.9
5		567		Edward		2.9
6		678		Fay		3.8
7		789		Gary		3.4
8		987		Helen		3.7
9		876		Irene		3.9
10		765		Jay		2.9
11		654		Amy		3.9
12		543		Craig		3.4

SQL

```
SELECT    SID, sName, GPA
FROM      Student
WHERE     GPA > 3.5 OR
          GPA <= 3.5 OR
          GPA IS NULL;
```

Result

	RZ	SID	RZ	SNAME	RZ	GPA
1		123		Amy		3.9
2		234		Bob		3.6
3		345		Craig		3.5
4		456		Doris		3.9
5		567		Edward		2.9
6		678		Fay		3.8
7		789		Gary		3.4
8		987		Helen		3.7
9		876		Irene		3.9
10		765		Jay		2.9
11		654		Amy		3.9
12		543		Craig		3.4
13		432		Kevin		(null)
14		321		Lori		(null)

SQL

```
SELECT    SID, sName,  
          GPA, sizeHS  
FROM      Student  
WHERE     GPA > 3.5 OR  
          sizeHS < 1600;
```

Result

	PK	SID	PK	SNAME	PK	GPA	PK	SIZEHS
1		123		Amy		3.9		1000
2		234		Bob		3.6		1500
3		345		Craig		3.5		500
4		456		Doris		3.9		1000
5		678		Fay		3.8		200
6		789		Gary		3.4		800
7		987		Helen		3.7		800
8		876		Irene		3.9		400
9		765		Jay		2.9		1500
10		654		Amy		3.9		1000
11		432		Kevin		(null)		1500

SQL

```
SELECT    SID, sName,  
          GPA, sizeHS  
FROM      Student  
WHERE     GPA > 3.5 OR  
          sizeHS < 1600 OR  
          sizeHS >= 1600;
```

Result

	PK	SID	PK	SNAME	PK	GPA	PK	SIZEHS
1		123		Amy		3.9		1000
2		234		Bob		3.6		1500
3		345		Craig		3.5		500
4		456		Doris		3.9		1000
5		567		Edward		2.9		2000
6		678		Fay		3.8		200
7		789		Gary		3.4		800
8		987		Helen		3.7		800
9		876		Irene		3.9		400
10		765		Jay		2.9		1500
11		654		Amy		3.9		1000
12		543		Craig		3.4		2000
13		432		Kevin		(null)		1500
14		321		Lori		(null)		2500

SQL

```
SELECT    COUNT(*)  
FROM      Student  
WHERE     GPA IS NOT NULL;
```

Result

	COUNT(*)
1	12

SQL

```
SELECT    COUNT(DISTINCT GPA)
FROM      Student
WHERE     GPA IS NOT NULL;
```

Result

	COUNT(DISTINCTGPA)
1	7

SQL

```
SELECT DISTINCT GPA  
FROM Student;
```

Result

	R2	GPA
1		(null)
2		3.6
3		3.7
4		3.5
5		2.9
6		3.8
7		3.9
8		3.4

Data Modification Statements

- Inserting new data (2 methods)

```
INSERT INTO TABLE  
VALUES (A1, A2, ..., An)
```

```
INSERT INTO TABLE  
Select-Statement
```

Data Modification Statements

- Deleting existing data

```
DELETE FROM TABLE  
WHERE Condition
```

Data Modification Statements

- Updating existing data

```
UPDATE TABLE  
SET Attribute = Expression  
WHERE Condition
```

Data Modification Statements

- Updating existing data

```
UPDATE TABLE  
SET A1 = Expr1, A2 = Expr2, ...,  
    A3 = Expr3,  
WHERE Condition
```

Demo: simple college admissions database

College(cName, state, enrollment)

Student(sID, sName, GPA, sizeHS)

Apply(sID, cName, major, decision)

SQL

```
INSERT INTO college
VALUES ('Carnegie Mellon',
       'PA',
       11500);
```

```
SELECT *
FROM   college;
```

Result

	 CNAME	 STATE	 ENROLLMENT
1	Stanford	CA	15000
2	Berkeley	CA	36000
3	MIT	MA	10000
4	Cornell	NY	21000
5	Carnegie Mellon	PA	11500

SQL

```
SELECT *
FROM Student
WHERE SID NOT IN
(
    SELECT SID
    FROM Apply
);
```

Result

	SID	SNAME	GPA	SIZEHS
1	456	Doris	3.9	1000
2	567	Edward	2.9	2000
3	789	Gary	3.4	800
4	654	Amy	3.9	1000

SQL

```
SELECT  sID, 'Carnegie Mellon',  
        'CS', NULL  
FROM    Student  
WHERE   sID NOT IN  
(  
    SELECT  sID  
    FROM    Apply  
);
```

Result

	R2	SID	R2	'CARNEGIE MELLON'	R2	'CS'	R2	NULL
1		456		Carnegie Mellon		CS		(null)
2		567		Carnegie Mellon		CS		(null)
3		789		Carnegie Mellon		CS		(null)
4		654		Carnegie Mellon		CS		(null)

SQL

```
INSERT INTO Apply
SELECT  SID, 'Carnegie Mellon',
        'CS', NULL
FROM    Student
WHERE   SID NOT IN
(
    SELECT  SID
    FROM    Apply
);
```

```
SELECT  *
FROM    Apply;
```

Result

	A Z SID	A Z CNAME	A Z MAJOR	A Z DECISION
1	123	Stanford	CS	Y
2	123	Stanford	EE	N
3	123	Berkeley	CS	Y
4	123	Cornell	EE	Y
5	234	Berkeley	biology	N
6	345	MIT	bioengineering	Y
7	345	Cornell	bioengineering	N
8	345	Cornell	CS	Y
9	345	Cornell	EE	N
10	678	Stanford	history	Y
11	987	Stanford	CS	Y
12	987	Berkeley	CS	Y
13	876	Stanford	CS	N
14	876	MIT	biology	Y
15	876	MIT	marine biology	N
16	765	Stanford	history	Y
17	765	Cornell	history	N
18	765	Cornell	psychology	Y
19	543	MIT	CS	N
20	456	Carnegie Mellon	CS	(null)
21	567	Carnegie Mellon	CS	(null)
22	789	Carnegie Mellon	CS	(null)
23	654	Carnegie Mellon	CS	(null)

SQL

```
SELECT *
FROM Student
WHERE SID IN
(
    SELECT SID
    FROM Apply
    WHERE major = 'EE'
    AND decision = 'N'
);
```

Result

	SID	SNAME	GPA	SIZEHS
1	123	Amy	3.9	1000
2	345	Craig	3.5	500

SQL

```
SELECT  SID, 'Carnegie Mellon',  
        'EE', 'Y'  
FROM    Student  
WHERE   SID IN  
(  
    SELECT  SID  
    FROM    Apply  
    WHERE   major = 'EE'  
    AND     decision = 'N'  
);
```

Result

	SID	'CARNEGIE MELLON'	'EE'	'Y'
1	123	Carnegie Mellon	EE	Y
2	345	Carnegie Mellon	EE	Y

SQL

```
INSERT INTO Apply
SELECT  SID, 'Carnegie Mellon',
        'EE', 'Y'
FROM    Student
WHERE   SID IN
(
    SELECT  SID
    FROM    Apply
    WHERE   major = 'EE'
    AND     decision = 'N'
);
```

```
SELECT  *
FROM    Apply;
```

Result

	A Z SID	A Z CNAME	A Z MAJOR	A Z DECISION
1	123	Stanford	CS	Y
2	123	Stanford	EE	N
3	123	Berkeley	CS	Y
4	123	Cornell	EE	Y
5	234	Berkeley	biology	N
6	345	MIT	bioengineering	Y
7	345	Cornell	bioengineering	N
8	345	Cornell	CS	Y
9	345	Cornell	EE	N
10	678	Stanford	history	Y
11	987	Stanford	CS	Y
12	987	Berkeley	CS	Y
13	876	Stanford	CS	N
14	876	MIT	biology	Y
15	876	MIT	marine biology	N
16	765	Stanford	history	Y
17	765	Cornell	history	N
18	765	Cornell	psychology	Y
19	543	MIT	CS	N
20	456	Carnegie Mellon	CS	(null)
21	567	Carnegie Mellon	CS	(null)
22	789	Carnegie Mellon	CS	(null)
23	654	Carnegie Mellon	CS	(null)
24	123	Carnegie Mellon	EE	Y
25	345	Carnegie Mellon	EE	Y

SQL

```
SELECT      SID,
            COUNT(DISTINCT major)
FROM        Apply
GROUP BY    SID
HAVING      COUNT(DISTINCT major)
            > 2;
```

Result

	RANK	SID	RANK	COUNT(DISTINCTMAJOR)
	1	876		3
	2	345		3

SQL

```
DELETE
FROM      Student
WHERE     SID IN
(
    SELECT      SID
    FROM        Apply
    GROUP BY    SID
    HAVING      COUNT(DISTINCT major)
                > 2
);
```

```
SELECT    *
FROM      Student;
```

Result

	PK	SID	PK	SNAME	PK	GPA	PK	SIZEHS
1		123		Amy		3.9		1000
2		234		Bob		3.6		1500
3		456		Doris		3.9		1000
4		567		Edward		2.9		2000
5		678		Fay		3.8		200
6		789		Gary		3.4		800
7		987		Helen		3.7		800
8		765		Jay		2.9		1500
9		654		Amy		3.9		1000
10		543		Craig		3.4		2000

SQL

```
DELETE
FROM      Apply
WHERE     SID IN
(
    SELECT  SID
    FROM    Apply
    GROUP BY SID
    HAVING  COUNT(DISTINCT major)
           > 2
);
```

```
SELECT  *
FROM    Apply;
```

Result

	SID	CNAME	MAJOR	DECISION
1	123	Stanford	CS	Y
2	123	Stanford	EE	N
3	123	Berkeley	CS	Y
4	123	Cornell	EE	Y
5	234	Berkeley	biology	N
6	678	Stanford	history	Y
7	987	Stanford	CS	Y
8	987	Berkeley	CS	Y
9	765	Stanford	history	Y
10	765	Cornell	history	N
11	765	Cornell	psychology	Y
12	543	MIT	CS	N
13	456	Carnegie Mellon	CS	(null)
14	567	Carnegie Mellon	CS	(null)
15	789	Carnegie Mellon	CS	(null)
16	654	Carnegie Mellon	CS	(null)
17	123	Carnegie Mellon	EE	Y

SQL

```
SELECT *  
FROM College  
WHERE cName NOT IN  
(  
    SELECT cName  
    FROM Apply  
    WHERE major = 'CS'  
);
```

Result

	 CNAME	 STATE	 ENROLLMENT
1	Cornell	NY	21000

SQL

```
DELETE
FROM    college
WHERE    cName NOT IN
(
    SELECT    cName
    FROM      Apply
    WHERE     major = 'CS'
);
```

```
SELECT    *
FROM      college;
```

Result

	 CNAME	 STATE	 ENROLLMENT
1	Stanford	CA	15000
2	Berkeley	CA	36000
3	MIT	MA	10000
4	Carnegie Mellon	PA	11500

SQL

```
SELECT *
FROM Apply
WHERE cName = 'Carnegie Mellon'
      AND sID IN
(
  SELECT sID
  FROM Student
  WHERE GPA < 3.6
);
```

Result

	RZ SID	RZ CNAME	RZ MAJOR	RZ DECISION
1	567	Carnegie Mellon	CS	(null)
2	789	Carnegie Mellon	CS	(null)

SQL

```
UPDATE Apply
SET    decision = 'Y',
       major = 'economics'
WHERE  cName =
        'Carnegie Mellon'
      AND SID IN
(
  SELECT SID
  FROM   Student
  WHERE  GPA < 3.6
);
```

Result

	PK SID	PK CNAME	PK MAJOR	PK DECISION
1	123	Stanford	CS	Y
2	123	Stanford	EE	N
3	123	Berkeley	CS	Y
4	123	Cornell	EE	Y
5	234	Berkeley	biology	N
6	678	Stanford	history	Y
7	987	Stanford	CS	Y
8	987	Berkeley	CS	Y
9	765	Stanford	history	Y
10	765	Cornell	history	N
11	765	Cornell	psychology	Y
12	543	MIT	CS	N
13	456	Carnegie Mellon	CS	(null)
14	567	Carnegie Mellon	economics	Y
15	789	Carnegie Mellon	economics	Y
16	654	Carnegie Mellon	CS	(null)
17	123	Carnegie Mellon	EE	Y

SQL

```
SELECT *
FROM Apply
WHERE major = 'EE'
      AND SID IN
( SELECT SID
  FROM Student
  WHERE GPA >= ALL
    ( SELECT GPA
      FROM Student
      WHERE SID IN
        ( SELECT SID
          FROM Apply
          WHERE major = 'EE'
        )
      )
    )
);
```

Result

	 SID	 CNAME	 MAJOR	 DECISION
1	123	Stanford	EE	N
2	123	Cornell	EE	Y
3	123	Carnegie Mellon	EE	Y

SQL

```
UPDATE Apply
SET major = 'CSE'
WHERE major = 'EE'
  AND SID IN
( SELECT SID
  FROM Student
 WHERE GPA >= ALL
  ( SELECT GPA
    FROM Student
   WHERE SID IN
    ( SELECT SID
      FROM Apply
     WHERE major = 'EE'
    )
  )
);
```

Result

	PK SID	PK CNAME	PK MAJOR	PK DECISION
1	123	Stanford	CS	Y
2	123	Stanford	CSE	N
3	123	Berkeley	CS	Y
4	123	Cornell	CSE	Y
5	234	Berkeley	biology	N
6	678	Stanford	history	Y
7	987	Stanford	CS	Y
8	987	Berkeley	CS	Y
9	765	Stanford	history	Y
10	765	Cornell	history	N
11	765	Cornell	psychology	Y
12	543	MIT	CS	N
13	456	Carnegie Mellon	CS	(null)
14	567	Carnegie Mellon	economics	Y
15	789	Carnegie Mellon	economics	Y
16	654	Carnegie Mellon	CS	(null)
17	123	Carnegie Mellon	CSE	Y

SQL

```
UPDATE Student
SET
GPA = ( SELECT MAX(GPA)
        FROM Student ),
sizeHS = ( SELECT MIN(sizeHS)
           FROM Student );
```

```
SELECT *
FROM Student;
```

Result

	PK SID	PK SNAME	PK GPA	PK SIZEHS
1	123	Amy	3.9	200
2	234	Bob	3.9	200
3	456	Doris	3.9	200
4	567	Edward	3.9	200
5	678	Fay	3.9	200
6	789	Gary	3.9	200
7	987	Helen	3.9	200
8	765	Jay	3.9	200
9	654	Amy	3.9	200
10	543	Craig	3.9	200

SQL

```
UPDATE Apply
SET decision = 'Y';
```

```
SELECT *
FROM Apply;
```

Result

	RA	SID	RA	CNAME	RA	MAJOR	RA	DECISION
1		123		Stanford		CS		Y
2		123		Stanford		CSE		Y
3		123		Berkeley		CS		Y
4		123		Cornell		CSE		Y
5		234		Berkeley		biology		Y
6		678		Stanford		history		Y
7		987		Stanford		CS		Y
8		987		Berkeley		CS		Y
9		765		Stanford		history		Y
10		765		Cornell		history		Y
11		765		Cornell		psychology		Y
12		543		MIT		CS		Y
13		456		Carnegie Mellon		CS		Y
14		567		Carnegie Mellon		economics		Y
15		789		Carnegie Mellon		economics		Y
16		654		Carnegie Mellon		CS		Y
17		123		Carnegie Mellon		CSE		Y

JOIN Operators

SELECT	$A_1, A_2, \dots, A_n$
FROM	$R_1, R_2, \dots, R_m$
WHERE	condition

Explicit **Join** of tables

JOIN Operators

SELECT
FROM
WHERE

$A_1, A_2, \dots, A_n$
 $R_1, R_2, \dots, R_m$
condition

Explicit Join of tables

- Inner Join On *Condition* $\longrightarrow \bowtie_{\Theta}$

JOIN Operators

SELECT
FROM
WHERE

$A_1, A_2, \dots, A_n$
 $R_1, R_2, \dots, R_m$
condition

Explicit Join of tables

- Inner Join On *Condition* $\longrightarrow \bowtie_{\Theta}$
- Natural Join $\longrightarrow \bowtie$

JOIN Operators

SELECT
FROM
WHERE

$A_1, A_2, \dots, A_n$
 $R_1, R_2, \dots, R_m$
condition

Explicit Join of tables

- Inner Join On *Condition* $\longrightarrow \bowtie_{\Theta}$
- Natural Join $\longrightarrow \bowtie$
- Inner Join Using(*attributes*) $\longrightarrow \bowtie$

JOIN Operators

SELECT
FROM
WHERE

$A_1, A_2, \dots, A_n$
 $R_1, R_2, \dots, R_m$
condition

Explicit Join of tables

- Inner Join On *Condition* $\longrightarrow \bowtie_{\Theta}$
- Natural Join $\longrightarrow \bowtie$
- Inner Join Using(*attributes*) $\longrightarrow \bowtie$
- Left | Right | Full Outer Join

SQL

```
SELECT DISTINCT sName, major
FROM Student, Apply
WHERE Student.sID = Apply.sID;
```

Result

	 SNAME	 MAJOR
1	Amy	EE
2	Bob	biology
3	Craig	EE
4	Irene	biology
5	Jay	psychology
6	Amy	CS
7	Craig	CS
8	Jay	history
9	Craig	bioengineering
10	Fay	history
11	Irene	marine biology
12	Helen	CS
13	Irene	CS

SQL

```
SELECT DISTINCT sName, major
FROM Student INNER JOIN Apply
ON Student.sID = Apply.sID;
```

Result

	 SNAME	 MAJOR
1	Amy	EE
2	Bob	biology
3	Craig	EE
4	Irene	biology
5	Jay	psychology
6	Amy	CS
7	Craig	CS
8	Jay	history
9	Craig	bioengineering
10	Fay	history
11	Irene	marine biology
12	Helen	CS
13	Irene	CS

SQL

```
SELECT DISTINCT sName, major
FROM Student JOIN Apply
ON Student.sID = Apply.sID;
```

Result

	SNAME	MAJOR
1	Amy	EE
2	Bob	biology
3	Craig	EE
4	Irene	biology
5	Jay	psychology
6	Amy	CS
7	Craig	CS
8	Jay	history
9	Craig	bioengineering
10	Fay	history
11	Irene	marine biology
12	Helen	CS
13	Irene	CS

SQL

```
SELECT  sName, GPA
FROM    Student, Apply
WHERE   Student.sID = Apply.sID
AND     sizeHS < 1000
AND     major = 'CS'
AND     cName = 'Stanford';
```

Result

	R Z	SNAME	R Z	GPA
	1	Helen		3.7
	2	Irene		3.9

SQL

```
SELECT  sName, GPA
FROM    Student JOIN Apply
ON      Student.sID = Apply.sID
WHERE   sizeHS < 1000
AND     major = 'CS'
AND     cName = 'Stanford';
```

Result

	R Z SNAME	R Z GPA
1	Helen	3.7
2	Irene	3.9

SQL

```
SELECT  sName, GPA
FROM    Student JOIN Apply
ON      Student.sID = Apply.sID
AND     sizeHS < 1000
AND     major = 'CS'
AND     cName = 'Stanford';
```

Result

	R Z SNAME	R Z GPA
1	Helen	3.7
2	Irene	3.9

SQL

```
SELECT  sName, GPA
FROM    Student JOIN Apply
ON      Student.sID = Apply.sID
WHERE   sizeHS < 1000
AND     major = 'CS'
AND     cName = 'Stanford';
```

Result

	R Z SNAME	R Z GPA
1	Helen	3.7
2	Irene	3.9

SQL

```
SELECT Apply.sID, sName, GPA,  
        Apply.cName, enrollment  
FROM Apply, Student, College  
WHERE Apply.sID = Student.sID  
AND Apply.cName =  
      College.cName;
```

Result

	AZ	SID	AZ	SNAME	AZ	GPA	AZ	CNAME	AZ	ENROLLMENT
1		123		Amy		3.9		Cornell		21000
2		123		Amy		3.9		Berkeley		36000
3		123		Amy		3.9		Stanford		15000
4		123		Amy		3.9		Stanford		15000
5		234		Bob		3.6		Berkeley		36000
6		345		Craig		3.5		Cornell		21000
7		345		Craig		3.5		Cornell		21000
8		345		Craig		3.5		Cornell		21000
9		345		Craig		3.5		MIT		10000
10		678		Fay		3.8		Stanford		15000
11		987		Helen		3.7		Berkeley		36000
12		987		Helen		3.7		Stanford		15000
13		876		Irene		3.9		MIT		10000
14		876		Irene		3.9		MIT		10000
15		876		Irene		3.9		Stanford		15000
16		765		Jay		2.9		Cornell		21000
17		765		Jay		2.9		Cornell		21000
18		765		Jay		2.9		Stanford		15000
19		543		Craig		3.4		MIT		10000

SQL

```
SELECT Apply.sID, sName, GPA,  
        Apply.cName, enrollment  
FROM Apply JOIN Student  
        JOIN College  
ON Apply.sID = Student.sID  
AND Apply.cName =  
    College.cName;
```

Result

```
ORA-00905: 누락된 키워드  
00905. 00000 - "missing keyword"  
*Cause:  
*Action:  
1,067행, 35열에서 오류 발생
```

SQL

```
SELECT Apply.sID, sName, GPA,  
        Apply.cName, enrollment  
FROM    (Apply JOIN Student ON  
        Apply.sID = Student.sID)  
JOIN    College  
ON      Apply.cName =  
        College.cName;
```


SQL

```
SELECT Apply.sID, sName, GPA,  
        Apply.cName, enrollment  
FROM    (Apply JOIN Student ON  
        Apply.sID = Student.sID)  
JOIN    College  
ON      Apply.cName =  
        College.cName;
```

Result

	RZ	SID	RZ	SNAME	RZ	GPA	RZ	CNAME	RZ	ENROLLMENT
1		123		Amy		3.9		Cornell		21000
2		123		Amy		3.9		Berkeley		36000
3		123		Amy		3.9		Stanford		15000
4		123		Amy		3.9		Stanford		15000
5		234		Bob		3.6		Berkeley		36000
6		345		Craig		3.5		Cornell		21000
7		345		Craig		3.5		Cornell		21000
8		345		Craig		3.5		Cornell		21000
9		345		Craig		3.5		MIT		10000
10		678		Fay		3.8		Stanford		15000
11		987		Helen		3.7		Berkeley		36000
12		987		Helen		3.7		Stanford		15000
13		876		Irene		3.9		MIT		10000
14		876		Irene		3.9		MIT		10000
15		876		Irene		3.9		Stanford		15000
16		765		Jay		2.9		Cornell		21000
17		765		Jay		2.9		Cornell		21000
18		765		Jay		2.9		Stanford		15000
19		543		Craig		3.4		MIT		10000

SQL

```
SELECT DISTINCT sName, major
FROM      Student INNER JOIN Apply
ON        Student.sID = Apply.sID;
```

Result

	 SNAME	 MAJOR
1	Amy	EE
2	Bob	biology
3	Craig	EE
4	Irene	biology
5	Jay	psychology
6	Amy	CS
7	Craig	CS
8	Jay	history
9	Craig	bioengineering
10	Fay	history
11	Irene	marine biology
12	Helen	CS
13	Irene	CS

SQL

```
SELECT DISTINCT sName, major
FROM      Student
NATURAL JOIN Apply;
```

Result

	 SNAME	 MAJOR
1	Amy	EE
2	Bob	biology
3	Craig	EE
4	Irene	biology
5	Jay	psychology
6	Amy	CS
7	Craig	CS
8	Jay	history
9	Craig	bioengineering
10	Fay	history
11	Irene	marine biology
12	Helen	CS
13	Irene	CS

SQL

```
SELECT *  
FROM Student  
NATURAL JOIN Apply;
```

Result

	AZ	SID	AZ	SNAME	AZ	GPA	AZ	SIZEHS	AZ	CNAME	AZ	MAJOR	AZ	DECISION
1		123		Amy		3.9		1000		Cornell		EE		Y
2		123		Amy		3.9		1000		Berkeley		CS		Y
3		123		Amy		3.9		1000		Stanford		EE		N
4		123		Amy		3.9		1000		Stanford		CS		Y
5		234		Bob		3.6		1500		Berkeley		biology		N
6		345		Craig		3.5		500		Cornell		EE		N
7		345		Craig		3.5		500		Cornell		CS		Y
8		345		Craig		3.5		500		Cornell		bioengineering		N
9		345		Craig		3.5		500		MIT		bioengineering		Y
10		678		Fay		3.8		200		Stanford		history		Y
11		987		Helen		3.7		800		Berkeley		CS		Y
12		987		Helen		3.7		800		Stanford		CS		Y
13		876		Irene		3.9		400		MIT		marine biology		N
14		876		Irene		3.9		400		MIT		biology		Y
15		876		Irene		3.9		400		Stanford		CS		N
16		765		Jay		2.9		1500		Cornell		psychology		Y
17		765		Jay		2.9		1500		Cornell		history		N
18		765		Jay		2.9		1500		Stanford		history		Y
19		543		Craig		3.4		2000		MIT		CS		N

SQL

```
SELECT DISTINCT SID  
FROM      Student  
NATURAL JOIN Apply;
```

Result

	RZ	SID
1		123
2		678
3		543
4		987
5		876
6		234
7		345
8		765

SQL

```
SELECT DISTINCT SID  
FROM      Student, Apply;
```

Result

```
ORA-00918: 열의 정의가 애매합니다  
00918. 00000 - "column ambiguously defined"  
*Cause:  
*Action:  
1,087행, 17열에서 오류 발생
```

SQL

```
SELECT  sName, GPA
FROM    Student JOIN Apply
ON      Student.sID = Apply.sID
WHERE   sizeHS < 1000
AND     major = 'CS'
AND     cName = 'Stanford';
```

Result

	 SNAME	 GPA
1	Helen	3.7
2	Irene	3.9

SQL

```
SELECT  sName, GPA
FROM    Student
NATURAL JOIN Apply
WHERE   sizeHS < 1000
AND     major = 'CS'
AND     cName = 'Stanford';
```

Result

	 SNAME	 GPA
1	Helen	3.7
2	Irene	3.9

SQL

```
SELECT  sName, GPA
FROM    Student
JOIN    Apply USING(sID)
WHERE   sizeHS < 1000
AND     major = 'CS'
AND     cName = 'Stanford';
```

SQL

```
SELECT  sName, GPA
FROM    Student
JOIN    Apply USING(sID)
WHERE   sizeHS < 1000
AND     major = 'CS'
AND     cName = 'Stanford';
```

Result

	R Z SNAME	R Z GPA
1	Helen	3.7
2	Irene	3.9

SQL

```
SELECT *  
FROM Student s1  
NATURAL JOIN Student s2;
```

Result

	SID	SNAME	GPA	SIZEHS
1	123	Amy	3.9	1000
2	234	Bob	3.6	1500
3	345	Craig	3.5	500
4	456	Doris	3.9	1000
5	567	Edward	2.9	2000
6	678	Fay	3.8	200
7	789	Gary	3.4	800
8	987	Helen	3.7	800
9	876	Irene	3.9	400
10	765	Jay	2.9	1500
11	654	Amy	3.9	1000
12	543	Craig	3.4	2000

SQL

```
SELECT *  
FROM Student;
```

Result

	SID	SNAME	GPA	SIZEHS
1	123	Amy	3.9	1000
2	234	Bob	3.6	1500
3	345	Craig	3.5	500
4	456	Doris	3.9	1000
5	567	Edward	2.9	2000
6	678	Fay	3.8	200
7	789	Gary	3.4	800
8	987	Helen	3.7	800
9	876	Irene	3.9	400
10	765	Jay	2.9	1500
11	654	Amy	3.9	1000
12	543	Craig	3.4	2000

SQL

```
SELECT  sName, sID, cName, major
FROM    Student
INNER JOIN Apply USING(sID);
```

Result

	 SNAME	 SID	 CNAME	 MAJOR
1	Amy	123	Cornell	EE
2	Amy	123	Berkeley	CS
3	Amy	123	Stanford	EE
4	Amy	123	Stanford	CS
5	Bob	234	Berkeley	biology
6	Craig	345	Cornell	EE
7	Craig	345	Cornell	CS
8	Craig	345	Cornell	bioengineering
9	Craig	345	MIT	bioengineering
10	Fay	678	Stanford	history
11	Helen	987	Berkeley	CS
12	Helen	987	Stanford	CS
13	Irene	876	MIT	marine biology
14	Irene	876	MIT	biology
15	Irene	876	Stanford	CS
16	Jay	765	Cornell	psychology
17	Jay	765	Cornell	history
18	Jay	765	Stanford	history
19	Craig	543	MIT	CS

SQL

```
SELECT  sName, sID, cName, major
FROM    Student
OUTER JOIN Apply USING(sID);
```

Result

	 SNAME	 SID	 CNAME	 MAJOR
1	Amy	123	Cornell	EE
2	Amy	123	Berkeley	CS
3	Amy	123	Stanford	EE
4	Amy	123	Stanford	CS
5	Bob	234	Berkeley	biology
6	Craig	345	Cornell	EE
7	Craig	345	Cornell	CS
8	Craig	345	Cornell	bioengineering
9	Craig	345	MIT	bioengineering
10	Fay	678	Stanford	history
11	Helen	987	Berkeley	CS
12	Helen	987	Stanford	CS
13	Irene	876	MIT	marine biology
14	Irene	876	MIT	biology
15	Irene	876	Stanford	CS
16	Jay	765	Cornell	psychology
17	Jay	765	Cornell	history
18	Jay	765	Stanford	history
19	Craig	543	MIT	CS

SQL

```
SELECT  sName, sID, cName, major
FROM    Student
LEFT OUTER JOIN Apply USING(sID);
```

Result

	A2 SNAME	A2 SID	A2 CNAME	A2 MAJOR
1	Amy	123	Stanford	CS
2	Amy	123	Stanford	EE
3	Amy	123	Berkeley	CS
4	Amy	123	Cornell	EE
5	Bob	234	Berkeley	biology
6	Craig	345	MIT	bioengineering
7	Craig	345	Cornell	bioengineering
8	Craig	345	Cornell	CS
9	Craig	345	Cornell	EE
10	Fay	678	Stanford	history
11	Helen	987	Stanford	CS
12	Helen	987	Berkeley	CS
13	Irene	876	Stanford	CS
14	Irene	876	MIT	biology
15	Irene	876	MIT	marine biology
16	Jay	765	Stanford	history
17	Jay	765	Cornell	history
18	Jay	765	Cornell	psychology
19	Craig	543	MIT	CS
20	Doris	456	(null)	(null)
21	Edward	567	(null)	(null)
22	Gary	789	(null)	(null)
23	Amy	654	(null)	(null)

SQL

```
SELECT  sName, sID, cName, major
FROM    Student
LEFT JOIN Apply USING(sID);
```

Result

	A2	SNAME	A2	SID	A2	CNAME	A2	MAJOR
1		Amy		123		Stanford		CS
2		Amy		123		Stanford		EE
3		Amy		123		Berkeley		CS
4		Amy		123		Cornell		EE
5		Bob		234		Berkeley		biology
6		Craig		345		MIT		bioengineering
7		Craig		345		Cornell		bioengineering
8		Craig		345		Cornell		CS
9		Craig		345		Cornell		EE
10		Fay		678		Stanford		history
11		Helen		987		Stanford		CS
12		Helen		987		Berkeley		CS
13		Irene		876		Stanford		CS
14		Irene		876		MIT		biology
15		Irene		876		MIT		marine biology
16		Jay		765		Stanford		history
17		Jay		765		Cornell		history
18		Jay		765		Cornell		psychology
19		Craig		543		MIT		CS
20		Doris		456		(null)		(null)
21		Edward		567		(null)		(null)
22		Gary		789		(null)		(null)
23		Amy		654		(null)		(null)

SQL

```
SELECT  sName, sID, cName, major
FROM    Student
NATURAL LEFT OUTER JOIN Apply;
```

Result

	A2 SNAME	A2 SID	A2 CNAME	A2 MAJOR
1	Amy	123	Stanford	CS
2	Amy	123	Stanford	EE
3	Amy	123	Berkeley	CS
4	Amy	123	Cornell	EE
5	Bob	234	Berkeley	biology
6	Craig	345	MIT	bioengineering
7	Craig	345	Cornell	bioengineering
8	Craig	345	Cornell	CS
9	Craig	345	Cornell	EE
10	Fay	678	Stanford	history
11	Helen	987	Stanford	CS
12	Helen	987	Berkeley	CS
13	Irene	876	Stanford	CS
14	Irene	876	MIT	biology
15	Irene	876	MIT	marine biology
16	Jay	765	Stanford	history
17	Jay	765	Cornell	history
18	Jay	765	Cornell	psychology
19	Craig	543	MIT	CS
20	Doris	456	(null)	(null)
21	Edward	567	(null)	(null)
22	Gary	789	(null)	(null)
23	Amy	654	(null)	(null)

SQL

```
SELECT  sName, Student.sID,  
        cName, major  
FROM    Student, Apply  
WHERE   Student.sID = Apply.sID  
UNION  
SELECT  sName, sID, NULL, NULL  
FROM    Student  
WHERE   sID NOT IN  
( SELECT sID FROM Apply );
```

Result

	A2 SNAME	A2 SID	A2 CNAME	A2 MAJOR
1	Amy	123	Berkeley	CS
2	Amy	123	Cornell	EE
3	Amy	123	Stanford	CS
4	Amy	123	Stanford	EE
5	Amy	654	(null)	(null)
6	Bob	234	Berkeley	biology
7	Craig	345	Cornell	CS
8	Craig	345	Cornell	EE
9	Craig	345	Cornell	bioengineering
10	Craig	345	MIT	bioengineering
11	Craig	543	MIT	CS
12	Doris	456	(null)	(null)
13	Edward	567	(null)	(null)
14	Fay	678	Stanford	history
15	Gary	789	(null)	(null)
16	Helen	987	Berkeley	CS
17	Helen	987	Stanford	CS
18	Irene	876	MIT	biology
19	Irene	876	MIT	marine biology
20	Irene	876	Stanford	CS
21	Jay	765	Cornell	history
22	Jay	765	Cornell	psychology
23	Jay	765	Stanford	history

SQL

```
SELECT  sName, sID, cName, major
FROM    Student
NATURAL LEFT OUTER JOIN Apply;
```

Result

	A2 SNAME	A2 SID	A2 CNAME	A2 MAJOR
1	Amy	123	Stanford	CS
2	Amy	123	Stanford	EE
3	Amy	123	Berkeley	CS
4	Amy	123	Cornell	EE
5	Bob	234	Berkeley	biology
6	Craig	345	MIT	bioengineering
7	Craig	345	Cornell	bioengineering
8	Craig	345	Cornell	CS
9	Craig	345	Cornell	EE
10	Fay	678	Stanford	history
11	Helen	987	Stanford	CS
12	Helen	987	Berkeley	CS
13	Irene	876	Stanford	CS
14	Irene	876	MIT	biology
15	Irene	876	MIT	marine biology
16	Jay	765	Stanford	history
17	Jay	765	Cornell	history
18	Jay	765	Cornell	psychology
19	Craig	543	MIT	CS
20	Doris	456	(null)	(null)
21	Edward	567	(null)	(null)
22	Gary	789	(null)	(null)
23	Amy	654	(null)	(null)

SQL

```
INSERT INTO Apply VALUES (321,  
'MIT', 'history', 'N');  
INSERT INTO Apply VALUES (321,  
'MIT', 'psychology', 'Y');
```

SQL

```
SELECT  sName, sID, cName, major
FROM    Apply
NATURAL LEFT OUTER JOIN Student;
```

Result

	A2	SNAME	A2	SID	A2	CNAME	A2	MAJOR
1	Amy		123	Cornell	EE			
2	Amy		123	Berkeley	CS			
3	Amy		123	Stanford	EE			
4	Amy		123	Stanford	CS			
5	Bob		234	Berkeley	biology			
6	Craig		345	Cornell	EE			
7	Craig		345	Cornell	CS			
8	Craig		345	Cornell	bioengineering			
9	Craig		345	MIT	bioengineering			
10	Fay		678	Stanford	history			
11	Helen		987	Berkeley	CS			
12	Helen		987	Stanford	CS			
13	Irene		876	MIT	marine biology			
14	Irene		876	MIT	biology			
15	Irene		876	Stanford	CS			
16	Jay		765	Cornell	psychology			
17	Jay		765	Cornell	history			
18	Jay		765	Stanford	history			
19	Craig		543	MIT	CS			
20	(null)		321	MIT	psychology			
21	(null)		321	MIT	history			

SQL

```
SELECT  sName, sID, cName, major
FROM    Student
NATURAL RIGHT OUTER JOIN Apply;
```

Result

	A2 SNAME	A2 SID	A2 CNAME	A2 MAJOR
1	Amy	123	Cornell	EE
2	Amy	123	Berkeley	CS
3	Amy	123	Stanford	EE
4	Amy	123	Stanford	CS
5	Bob	234	Berkeley	biology
6	Craig	345	Cornell	EE
7	Craig	345	Cornell	CS
8	Craig	345	Cornell	bioengineering
9	Craig	345	MIT	bioengineering
10	Fay	678	Stanford	history
11	Helen	987	Berkeley	CS
12	Helen	987	Stanford	CS
13	Irene	876	MIT	marine biology
14	Irene	876	MIT	biology
15	Irene	876	Stanford	CS
16	Jay	765	Cornell	psychology
17	Jay	765	Cornell	history
18	Jay	765	Stanford	history
19	Craig	543	MIT	CS
20	(null)	321	MIT	psychology
21	(null)	321	MIT	history

SQL

```
SELECT  sName, sID, cName, major
FROM    Student
FULL OUTER JOIN Apply USING(sID);
```

Result

	R2 SNAME	R2 SID	R2 CNAME	R2 MAJOR
1	Amy	123	Cornell	EE
2	Amy	123	Berkeley	CS
3	Amy	123	Stanford	EE
4	Amy	123	Stanford	CS
5	Bob	234	Berkeley	biology
6	Craig	345	Cornell	EE
7	Craig	345	Cornell	CS
8	Craig	345	Cornell	bioengineering
9	Craig	345	MIT	bioengineering
10	Doris	456	(null)	(null)
11	Edward	567	(null)	(null)
12	Fay	678	Stanford	history
13	Gary	789	(null)	(null)
14	Helen	987	Berkeley	CS
15	Helen	987	Stanford	CS
16	Irene	876	MIT	marine biology
17	Irene	876	MIT	biology
18	Irene	876	Stanford	CS
19	Jay	765	Cornell	psychology
20	Jay	765	Cornell	history
21	Jay	765	Stanford	history
22	Amy	654	(null)	(null)
23	Craig	543	MIT	CS
24	(null)	321	MIT	psychology
25	(null)	321	MIT	history

SQL

```
SELECT  sName, sID, cName, major
FROM    Student
LEFT OUTER JOIN Apply USING(sID)
UNION
SELECT  sName, sID, cName, major
FROM    Student
RIGHT OUTER JOIN Apply USING(sID)
```

Result

	R2 SNAME	R2 SID	R2 CNAME	R2 MAJOR
1	Amy	123	Berkeley	CS
2	Amy	123	Cornell	EE
3	Amy	123	Stanford	CS
4	Amy	123	Stanford	EE
5	Amy	654	(null)	(null)
6	Bob	234	Berkeley	biology
7	Craig	345	Cornell	CS
8	Craig	345	Cornell	EE
9	Craig	345	Cornell	bioengineering
10	Craig	345	MIT	bioengineering
11	Craig	543	MIT	CS
12	Doris	456	(null)	(null)
13	Edward	567	(null)	(null)
14	Fay	678	Stanford	history
15	Gary	789	(null)	(null)
16	Helen	987	Berkeley	CS
17	Helen	987	Stanford	CS
18	Irene	876	MIT	biology
19	Irene	876	MIT	marine biology
20	Irene	876	Stanford	CS
21	Jay	765	Cornell	history
22	Jay	765	Cornell	psychology
23	Jay	765	Stanford	history
24	(null)	321	MIT	history
25	(null)	321	MIT	psychology

SQL

```
SELECT  sName, Student.sID,
        cName, major
FROM    Student, Apply
WHERE   Student.sID = Apply.sID
UNION
SELECT  sName, sID, NULL, NULL
FROM    Student
WHERE   sID NOT IN
        ( SELECT sID FROM Apply )
UNION
SELECT  NULL, sID, cName, major
FROM    Apply
WHERE   sID NOT IN
        ( SELECT sID FROM Student );
```

Result

R 2	SNAME	R 2	SID	R 2	CNAME	R 2	MAJOR
1	Amy		123	Berkeley	CS		
2	Amy		123	Cornell	EE		
3	Amy		123	Stanford	CS		
4	Amy		123	Stanford	EE		
5	Amy		654	(null)	(null)		
6	Bob		234	Berkeley	biology		
7	Craig		345	Cornell	CS		
8	Craig		345	Cornell	EE		
9	Craig		345	Cornell	bioengineering		
10	Craig		345	MIT	bioengineering		
11	Craig		543	MIT	CS		
12	Doris		456	(null)	(null)		
13	Edward		567	(null)	(null)		
14	Fay		678	Stanford	history		
15	Gary		789	(null)	(null)		
16	Helen		987	Berkeley	CS		
17	Helen		987	Stanford	CS		
18	Irene		876	MIT	biology		
19	Irene		876	MIT	marine biology		
20	Irene		876	Stanford	CS		
21	Jay		765	Cornell	history		
22	Jay		765	Cornell	psychology		
23	Jay		765	Stanford	history		
24	(null)		321	MIT	history		
25	(null)		321	MIT	psychology		

