Query Formulation with SQL

CHAPTER 3

OUTLINE

- Background
- Getting started
- Joining tables
- Summarizing tables
- Reasoning tools
- Advanced problems
- Data manipulation statements

WHAT IS SQL?

- Structured Query Language (Sequel)
- Language for database
  - definition,
  - manipulation, and
  - control
- International standard
- Standalone and embedded usage
- Intergalactic database speak

SQL STATEMENTS

1. Definition:
   - CREATE TABLE, ALTER TABLE, CREATE VIEW, CREATE SCHEMA
2. Manipulation:
   - SELECT, INSERT, UPDATE, DELETE, COMMIT, ROLLBACK
3. Control:
   - GRANT, REVOKE, CREATE ASSERTION
4. Other statements:
   - SET , CREATE TRIGGER, CREATE DOMAIN

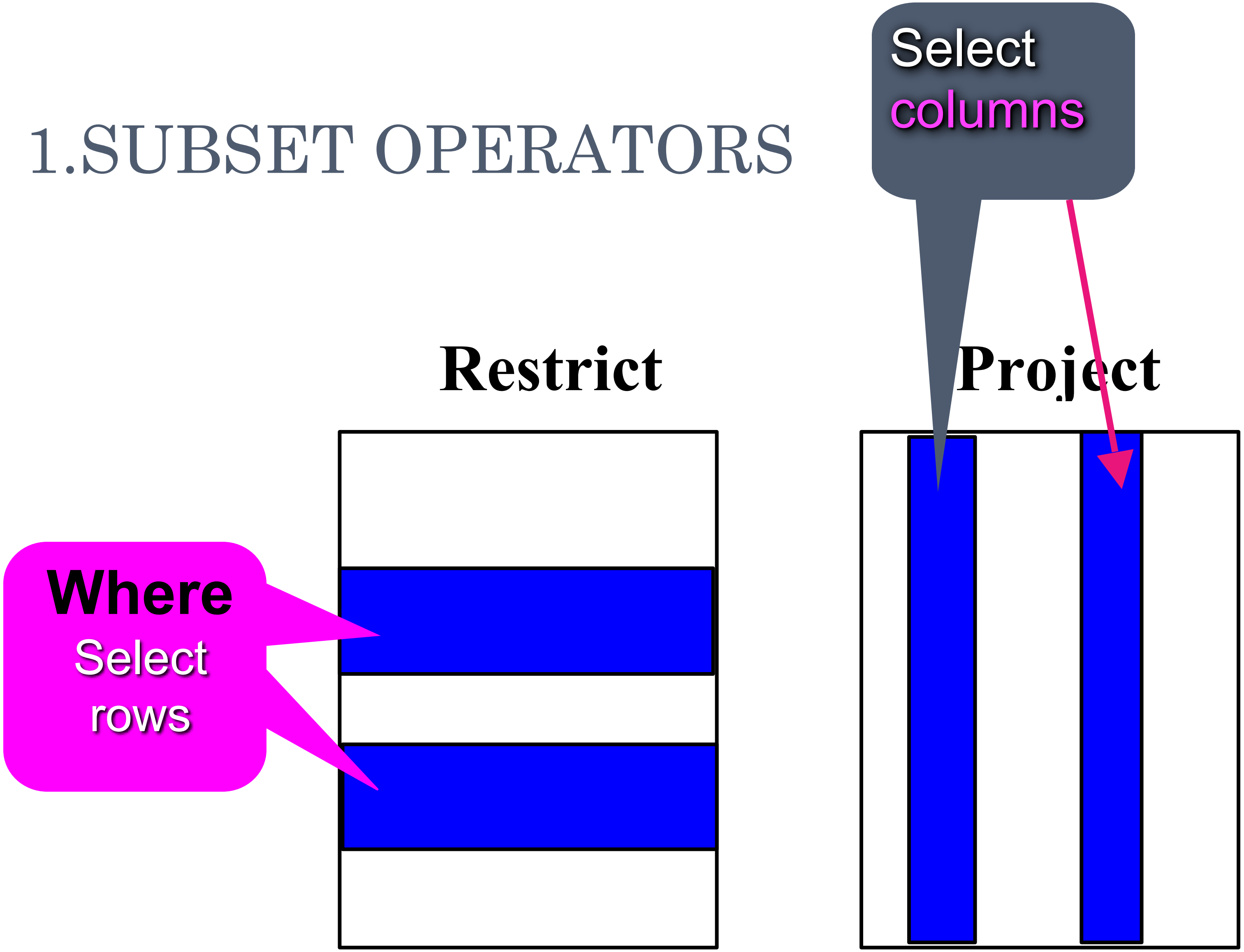
SQL STATEMENTS

Statement
CREATE TABLE
ALTER TABLE
SELECT
INSERT
UPDATE
DELETE
CREATE VIEW
CREATE TRIGGER
GRANT, REVOKE
CREATE ASSERTION
COMMIT, ROLLBACK

SELECT STATEMENT OVERVIEW

```
SELECT <columns,..>
  FROM <tables>
  WHERE <expressions for rows>
```

1.SUBSET OPERATORS



Select columns

Restrict

Project

Where

Select rows

NOTE : SQL COMMANDS

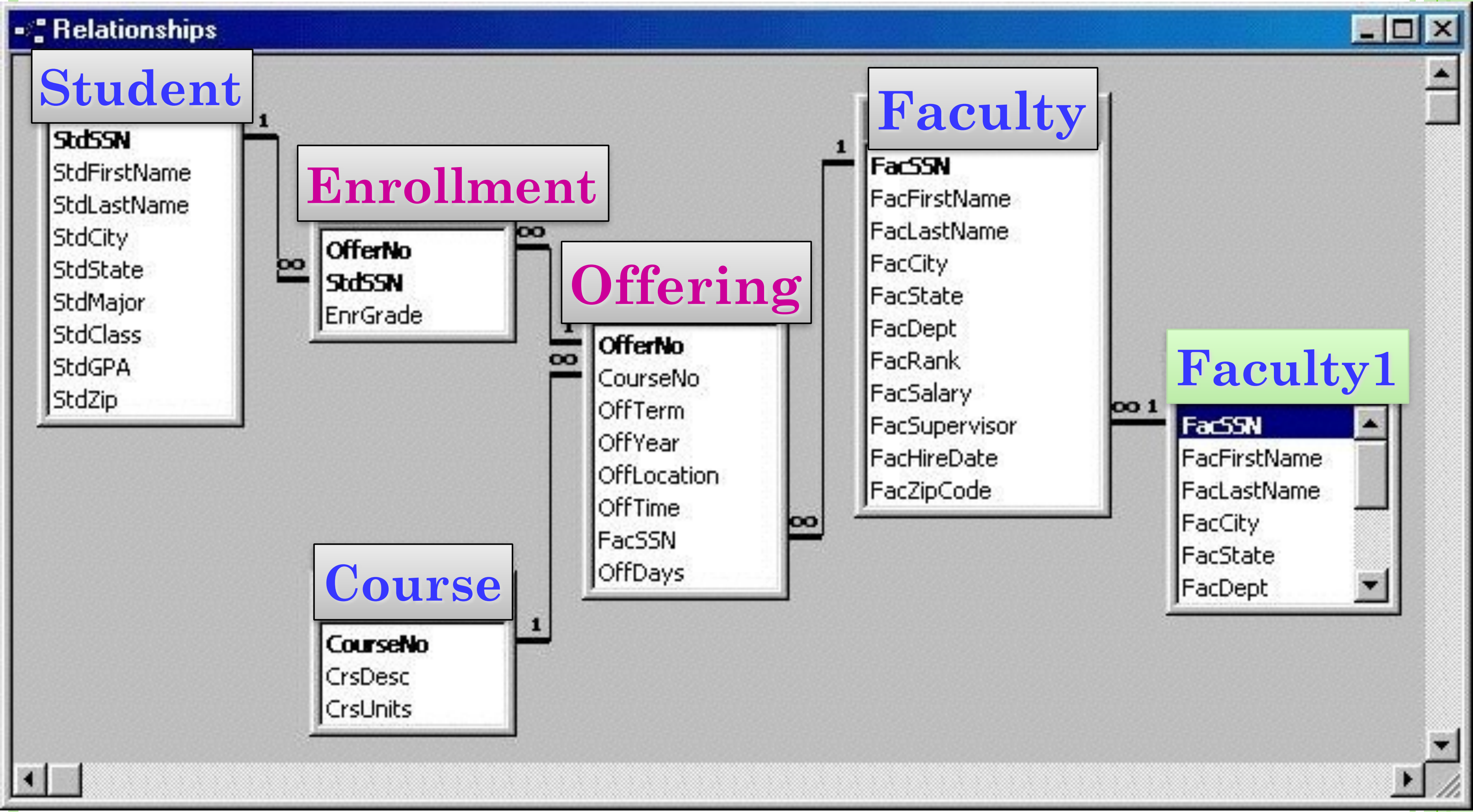
- **Meaning**
- **Conceptual Operation**
  - **step by step**

SELECT STATEMENT OVERVIEW

```
SELECT <list of column expressions>
  FROM <list of tables and join operations>
  WHERE <list of logical expressions for rows>
  GROUP BY <list of grouping columns>
  HAVING <list of logical expressions for groups>
  ORDER BY <list of sorting specifications>
```

- Expression: combination of columns, constants, operators, and functions

UNIVERSITY DATABASE



FIRST SELECT EXAMPLES

Example: select all rows and columns(*)

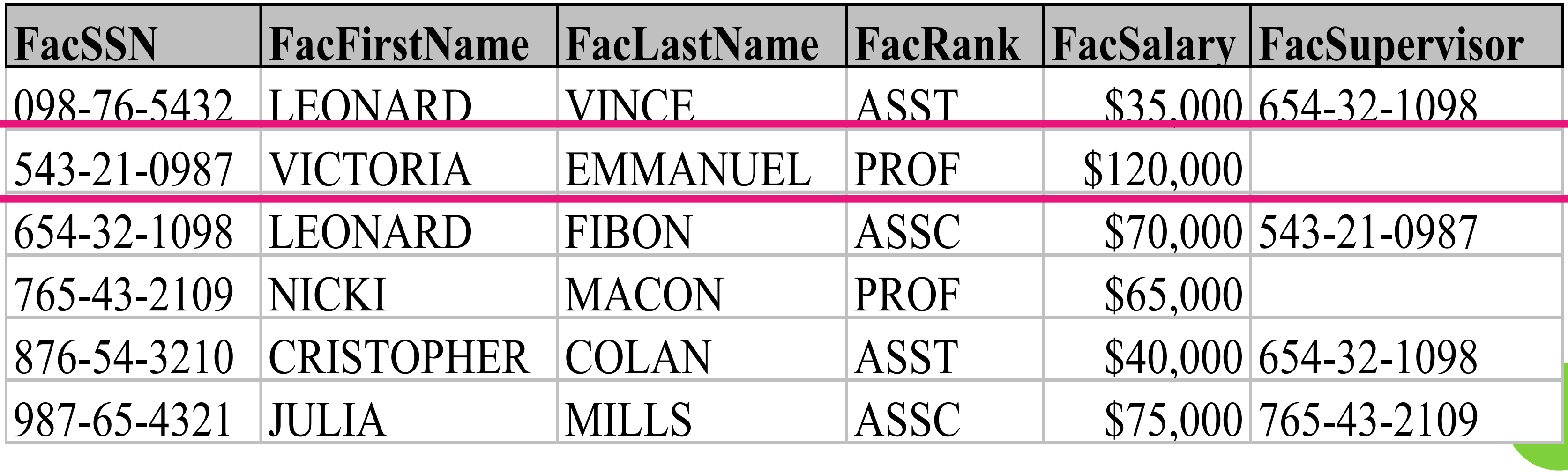
```
SELECT * FROM Faculty
```

FacSSN	FacFirstName	FacLastName	FacRank	FacSalary	FacSupervisor
098-76-5432	LEONARD	VINCE	ASST	$35,000	654-32-1098
543-21-0987	VICTORIA	EMMANUEL	PROF	$120,000	
654-32-1098	LEONARD	FIBON	ASSC	$70,000	543-21-0987
765-43-2109	NICKI	MACON	PROF	$65,000	
876-54-3210	CRISTOPHER	COLAN	ASST	$40,000	654-32-1098
987-65-4321	JULIA	MILLS	ASSC	$75,000	765-43-2109

FIRST SELECT EXAMPLES

Example 2 (Access) *select* all columns(*), *some rows,*

```
SELECT *
  FROM Faculty
  WHERE FacSSN = '543210987'
```

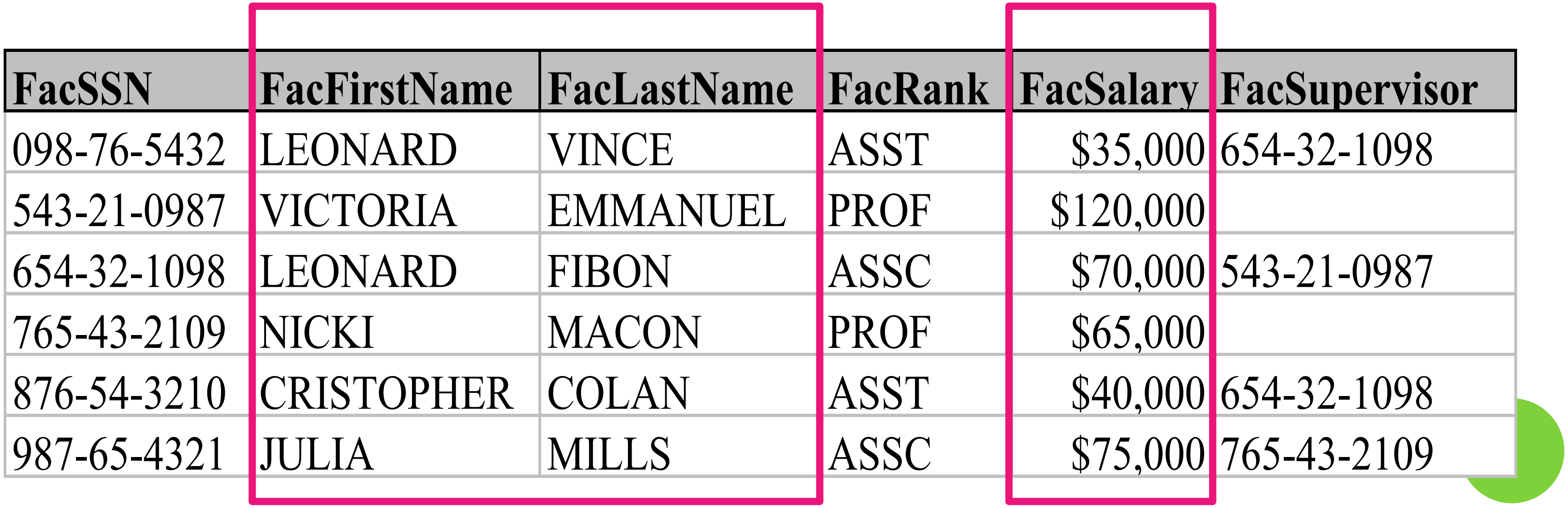


FacSSN	FacFirstName	FacLastName	FacRank	FacSalary	FacSupervisor
098-76-5432	LEONARD	VINCE	ASST	$35,000	654-32-1098
543-21-0987	VICTORIA	EMMANUEL	PROF	$120,000	
654-32-1098	LEONARD	FIBON	ASSC	$70,000	543-21-0987
765-43-2109	NICKI	MACON	PROF	$65,000	
876-54-3210	CRISTOPHER	COLAN	ASST	$40,000	654-32-1098
987-65-4321	JULIA	MILLS	ASSC	$75,000	765-43-2109

FIRST SELECT EXAMPLES

Example 3 select some columns and all rows

```
SELECT FacFirstName, FacLastName,
        FacSalary
  FROM Faculty
```



FacSSN	FacFirstName	FacLastName	FacRank	FacSalary	FacSupervisor
098-76-5432	LEONARD	VINCE	ASST	$35,000	654-32-1098
543-21-0987	VICTORIA	EMMANUEL	PROF	$120,000	
654-32-1098	LEONARD	FIBON	ASSC	$70,000	543-21-0987
765-43-2109	NICKI	MACON	PROF	$65,000	
876-54-3210	CRISTOPHER	COLAN	ASST	$40,000	654-32-1098
987-65-4321	JULIA	MILLS	ASSC	$75,000	765-43-2109

FIRST SELECT EXAMPLES

Example 4 select some columns and some rows

```
SELECT FacFirstName, FacLastName, FacSalary
  FROM Faculty
  WHERE FacSalary > 65000 AND FacRank = 'PROF'
```

FacSSN	FacFirstName	FacLastName	FacRank	FacSalary	FacSupervisor
098-76-5432	LEONARD	VINCE	ASST	$35,000	654-32-1098
543-21-0987	VICTORIA	EMMANUEL	PROF	$120,000	
654-32-1098	LEONARD	FIBON	ASSC	$70,000	543-21-0987
765-43-2109	NICKI	MACON	PROF	$65,000	
876-54-3210	CRISTOPHER	COLAN	ASST	$40,000	654-32-1098
987-65-4321	JULIA	MILLS	ASSC	$75,000	765-43-2109

USING EXPRESSIO

Retrieves faculty hired after 1991
- Inflates salary by 10%

Example 5 (Access)

```
SELECT FacFirstName, FacLastName,
       FacSalary*1.1 AS IncreasedSalary,
       FacHireDate
 FROM Faculty
 WHERE year(FacHireDate) > 1991
```

Multiply by 1.1

FacSSN	FacFirstName	FacLastName	FacRank	FacSalary	FacSupervisor	FacHireDate
098-76-5432	LEONARD	VINCE	ASST	$35,000	654-32-1098	01/01/1990
543-21-0987	VICTORIA	EMMANUEL	PROF	$120,000		01/05/1999
654-32-1098	LEONARD	FIBON	ASSC	$70,000	543-21-0987	01/02/2005
765-43-2109	NICKI	MACON	PROF	$65,000		01/02/2010
876-54-3210	CRISTOPHER	COLAN	ASST	$40,000	654-32-1098	01/02/2008
987-65-4321	JULIA	MILLS	ASSC	$75,000	765-43-2109	01/02/2009

USING EXPRESSIONS

Example 5 (Oracle)

```
SELECT FacFirstName, FacLastName, FacCity,
       FacSalary*1.1 AS IncreasedSalary,
       FacHireDate
 FROM Faculty
 WHERE to_number(to_char(FacHireDate,'YYYY'))
         > 1991
```

INEXACT MATCHING

- Match against a pattern: **LIKE operator**
- Use meta characters to specify patterns
  - Wildcard **(* or %)**
  - Any single character (? or _)

Example 6 (Access)

```
 SELECT *
  FROM Offering
  WHERE CourseNo LIKE '204*'
```

Example 6 (Oracle)…

```
WHERE CourseNo LIKE '204%'
```

USING DATES

- Dates are numbers
- Date constants and functions are not sta

Short cut for >= AND <=

Example 7 (Access)

```
SELECT FacFirstName, FacLastName, Fa HireDate
 FROM Faculty
 WHERE FacHireDate BETWEEN #1/1/1994#
   AND #12/31/1995#
```

Example 7 (Oracle)

```
SELECT FacFirstName, FacLastName, FacHireDate
 FROM Faculty
 WHERE FacHireDate BETWEEN '1-Jan-1994'
   AND '31-Dec-1995'
```

OTHER SINGLE TABLE EXAMPLES

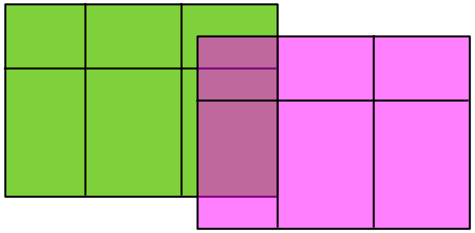
Example 8: Testing for null values

```
SELECT OfferNo, CourseNo
 FROM Offering
 WHERE FacSSN IS NULL AND OffTerm = 'SUMMER'
   AND OffYear = 2013
```

Example 9: Mixing AND and OR

```
SELECT OfferNo, CourseNo, FacSSN
 FROM Offering
 WHERE (OffTerm = 'FALL' AND OffYear = 2012)
    OR (OffTerm = 'WINTER' AND OffYear = 2013)
```

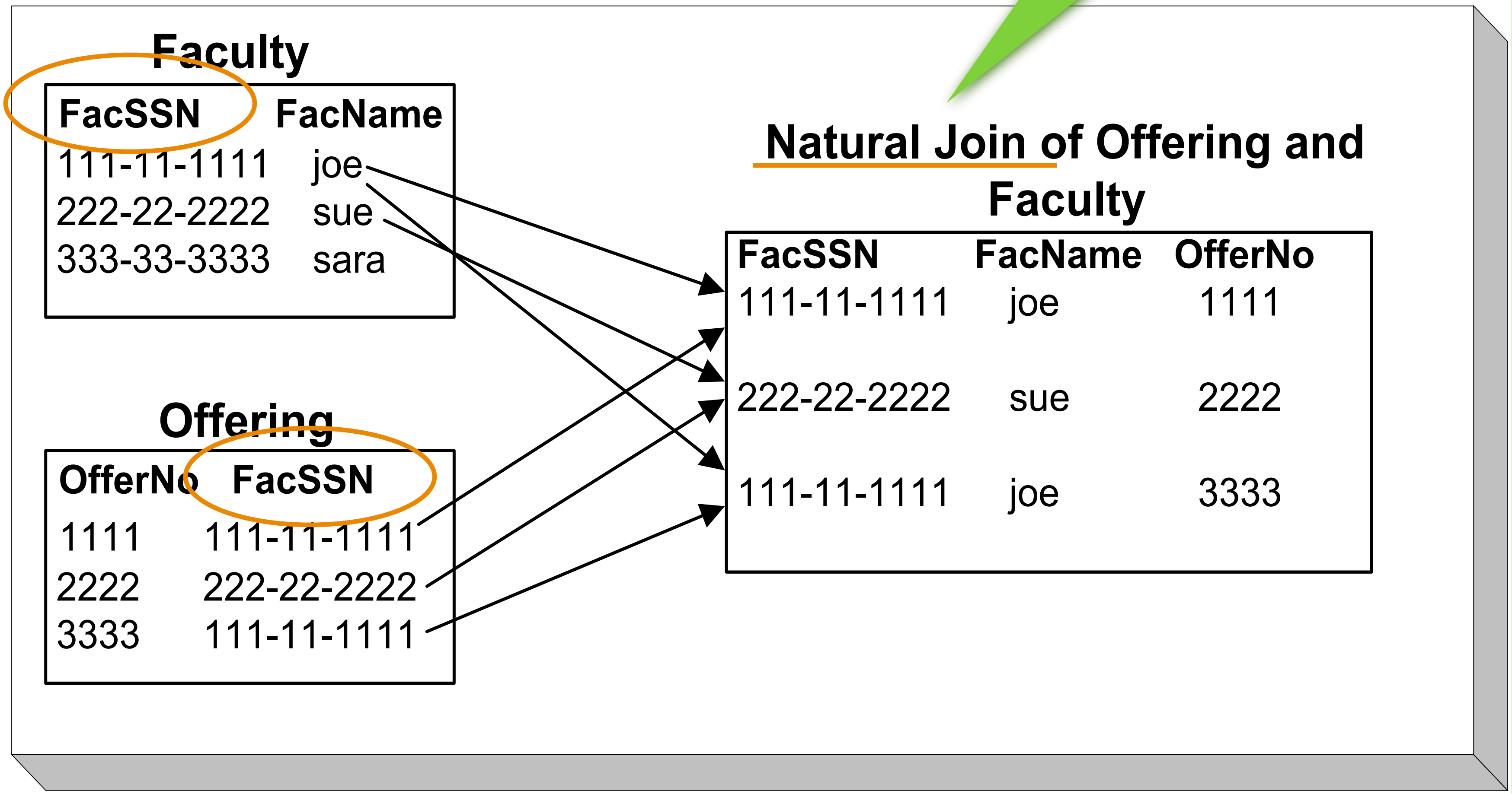
JOIN OPERATOR



- Most databases have **many tables**
- Combine tables using the join operator
- Specify matching condition
  - Can be any comparison but usually =
  - PK = FK most common join condition
  - Relationship diagram useful when combining tables

JOIN EXAMPLE

Join by FacSSN automatically



Faculty

FacSSN	FacName
111-11-1111	joe
222-22-2222	sue
333-33-3333	sara

Offering

OfferNo	FacSSN
1111	111-11-1111
2222	222-22-2222
3333	111-11-1111

Natural Join of Offering and Faculty

FacSSN	FacName	OfferNo
111-11-1111	joe	1111
222-22-2222	sue	2222
111-11-1111	joe	3333

CROSS PRODUCT STYLE

- List tables in the FROM clause
- List join conditions in the WHERE clause

Example 10 (Access)

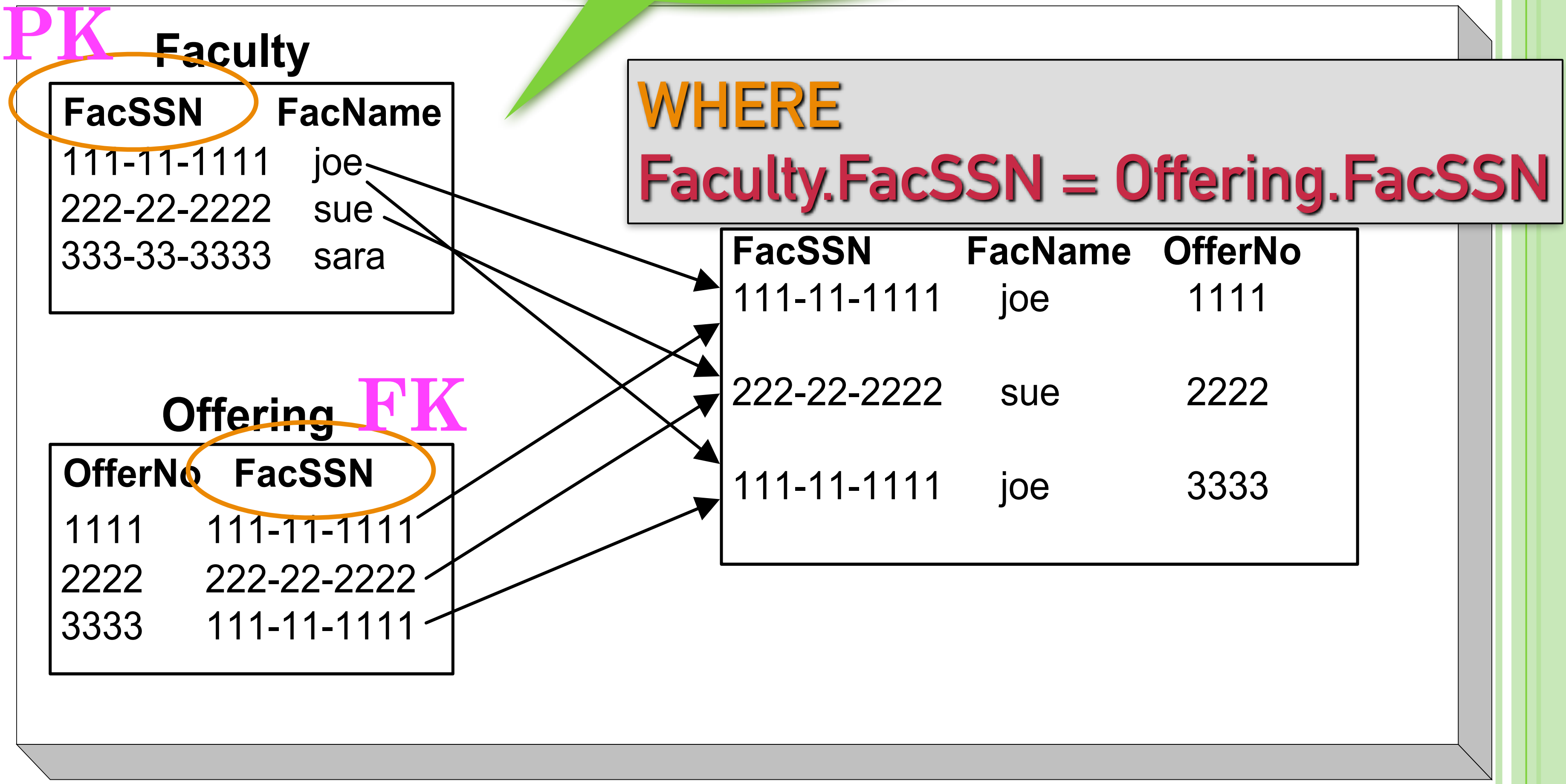
```
SELECT *
  FROM Offering, Faculty
```

No join condition

List all matched and unmatched rows of the JOIN

JOIN EXAM

Common Attributes: FacSSN



Faculty

FacSSN	FacName
111-11-1111	joe
222-22-2222	sue
333-33-3333	sara

Offering

OfferNo	FacSSN
1111	111-11-1111
2222	222-22-2222
3333	111-11-1111

WHERE Faculty.FacSSN = Offering.FacSSN

FacSSN	FacName	OfferNo
111-11-1111	joe	1111
222-22-2222	sue	2222
111-11-1111	joe	3333

Join Conditions

- List tables in the FROM clause
- List join conditions in the WHERE clause

Example 10.1 (Access)

```
SELECT OffYear,OffTerm,CourseNo,
       FacFirstName,FacLastName
FROM Offering, Faculty
WHERE Faculty.FacSSN = Offering.FacSSN
```

details of offerings and assigned faculty for all course offerings

CROSS PRODUCT STYL

Retrieve ?

Example 10.2 (Access)

```
SELECT OfferNo, CourseNo, FacFirstName,
       FacLastName
 FROM Offering, Faculty
 WHERE OffTerm = 'FALL' AND OffYear = 2012
   AND FacRank = 'ASST' AND CourseNo LIKE‘204*'
   AND Faculty.FacSSN = Offering.FacSSN
```

details of offerings and assigned faculty for fall 2012 IS courses taught by assistant professors

JOIN OPERATOR STYLE

- Use **INNER JOIN** and ON keywords
- **FROM clause contains JOIN operations**

Example 11 (Access)

```
SELECT OfferNo, CourseNo, FacFirstName,
       FacLastName
 FROM Offering INNER JOIN Faculty
   ON Faculty.FacSSN = Offering.FacSSN
 WHERE OffTerm = 'FALL' AND OffYear = 2012
   AND FacRank = 'ASST' AND CourseNo LIKE 'IS*'
```

JOIN OPERATOR STYLE

Equivalence of Example 11 (Access)

```
SELECT OfferNo, CourseNo, FacFirstName,
       FacLastName
FROM Offering,Faculty
WHERE OffTerm = 'FALL' AND OffYear = 2012
   AND FacRank = 'ASST' AND CourseNo LIKE 'IS*'
   AND Faculty.FacSSN = Offering.FacSSN
```

NAME QUALIFICATION

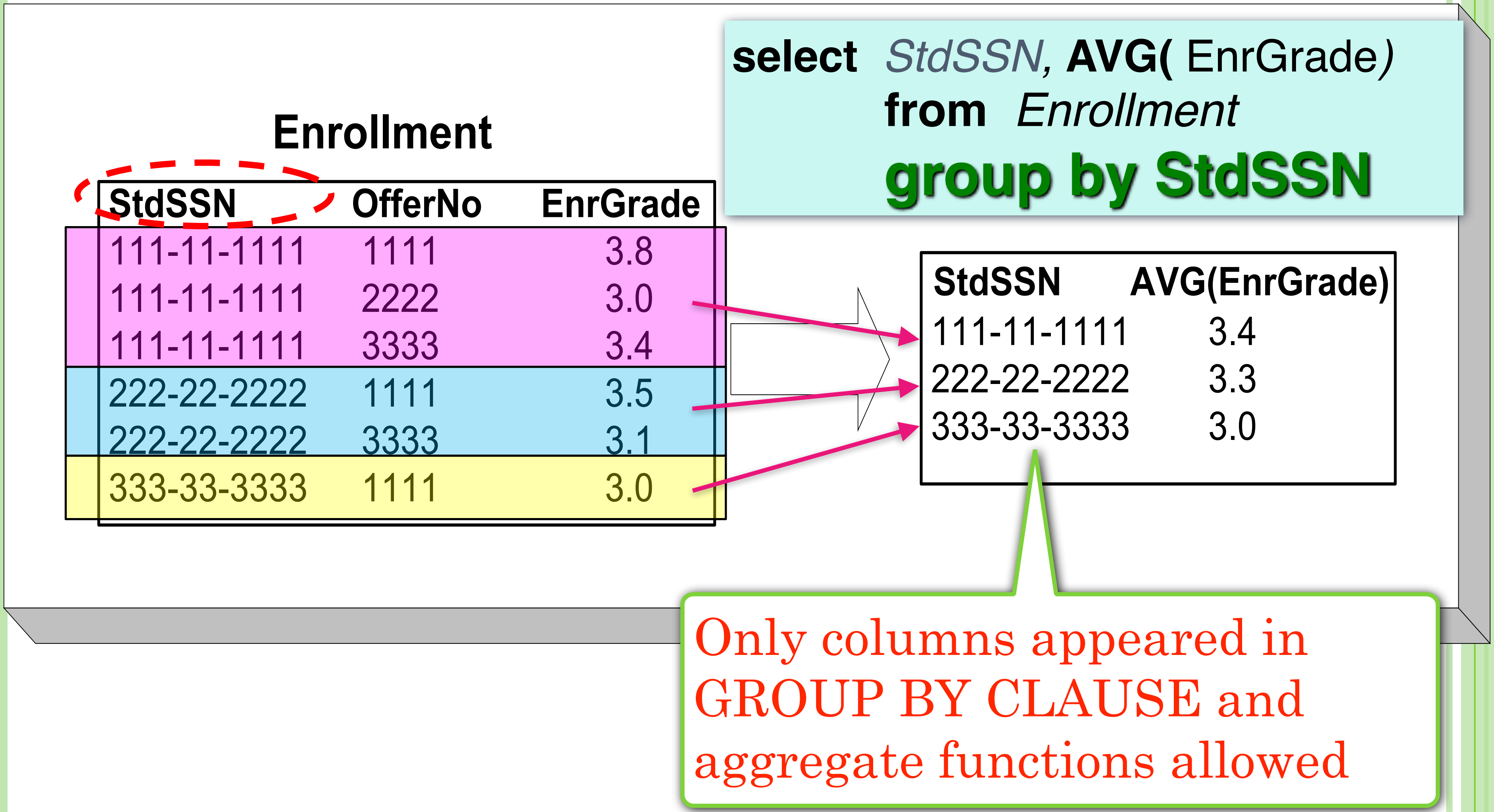
- Ambiguous column reference
  - More than one table in the query contains a column referenced in the query
  - Ambiguity determined by the query not the database
- Use column name alone if query is not ambiguous
- Qualify with table name if query is ambiguous

```
Faculty.FacSSN,
Offering.FacSSN
```

SUMMARIZING TABLES

- SQL keywords
  - Aggregate functions in the output list
  - GROUP BY: summary columns
  - HAVING: summary conditions

SUMMARIZE EXAMPLE



Enrollment

StdSSN	OfferNo	EnrGrade
111-11-1111	1111	3.8
111-11-1111	2222	3.0
111-11-1111	3333	3.4
222-22-2222	1111	3.5
222-22-2222	3333	3.1
333-33-3333	1111	3.0

```
select  StdSSN, AVG( EnrGrade)
        from  Enrollment
        group by StdSSN
```

StdSSN	AVG(EnrGrade)
111-11-1111	3.4
222-22-2222	3.3
333-33-3333	3.0

Only columns appeared in GROUP BY CLAUSE and aggregate functions allowed

GROUP BY EXAMPLES

Example 12: Grouping on a single column

```
SELECT FacRank,AVG(FacSalary)
                     AS AvgSalary
FROM Faculty
GROUP BY FacRank
```

GROUP BY EXAMPLES

Example 13: Row and group cond

Only columns appeared in GROUP BY CLAUSE and aggregate functions allowed

```
SELECT StdMajor, AVG(StdGPA) AS AvgGpa
  FROM Student
  WHERE StdClass IN ('JR', 'SR')
  GROUP BY StdMajor
  HAVING AVG(StdGPA) > 3.1
```

SQL SUMMARIZATION RULES

- Columns in SELECT and GROUP BY
  - SELECT: non aggregate and aggregate columns
  - GROUP BY: list <u>all</u> non aggregate columns
- WHERE versus HAVING
  - Row conditions in WHERE
  - Group conditions in HAVING

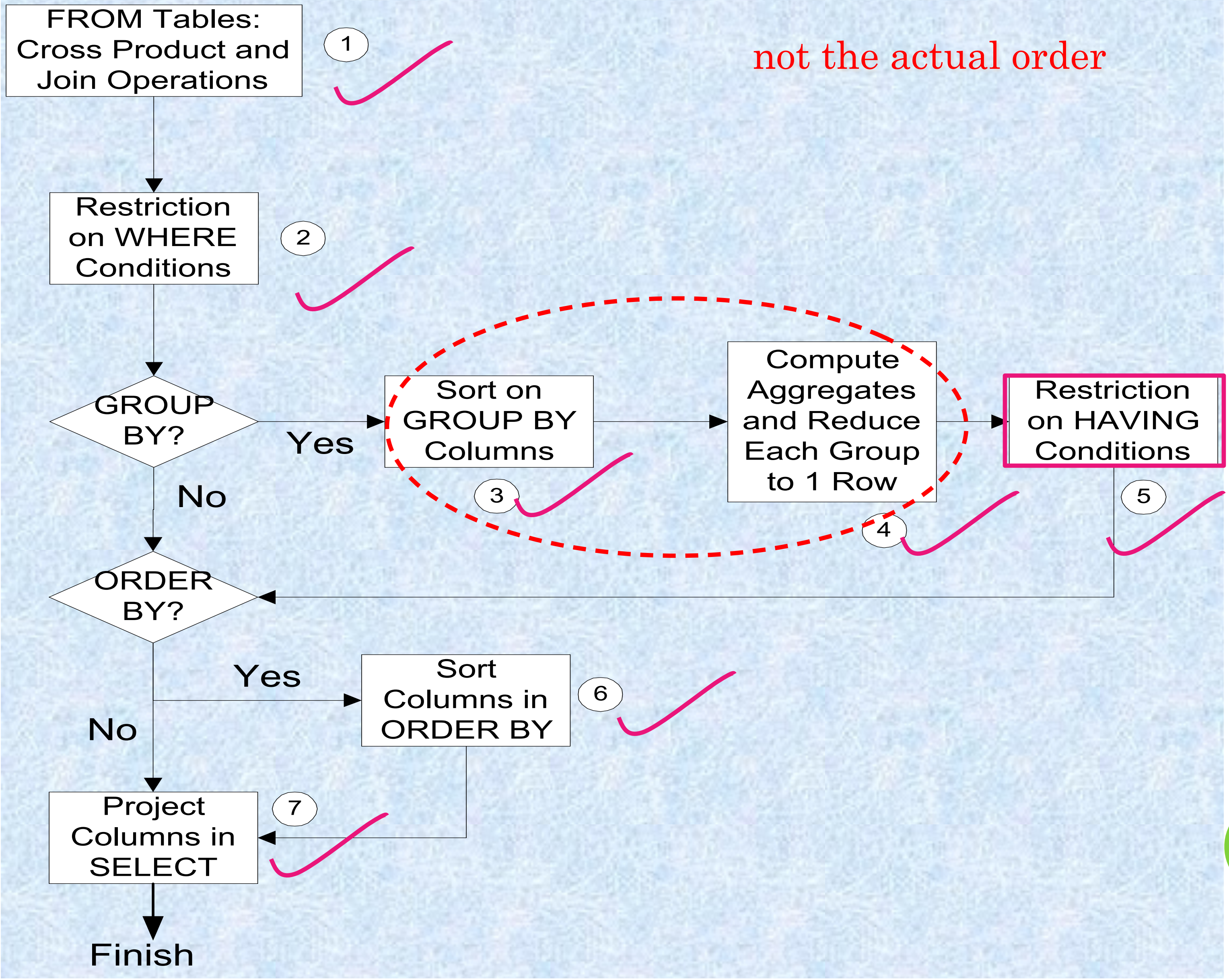
SUMMARIZATION AND JOINS

- Powerful combination
- List join conditions in the WHERE clause

Example 14: List the number of students enrolled in each fall 2013 offering.

```
SELECT Offering.OfferNo,
       COUNT(*) AS NumStudents
  FROM Enrollment, Offering
  WHERE Offering.OfferNo = Enrollment.OfferNo
    AND OffYear = 2013
  GROUP BY Offering.OfferNo
```

CONCEPTUAL EVALUATION PROCESS


FROM Tables: Cross Product and Join Operations
1
Restriction on WHERE Conditions
2
GROUP BY?
Yes
No
Sort on GROUP BY Columns
3
Compute Aggregates and Reduce Each Group to 1 Row
4
Restriction on HAVING Conditions
5
ORDER BY?
Yes
No
Sort Columns in ORDER BY
6
Project Columns in SELECT
7
Finish
not the actual order

CONCEPTUAL EVALUATION LESSONS

- **ROW operations** before GROUP operations
  - FROM and WHERE before GROUP BY and HAVING
  - Check row operations first
- Grouping occurs only one time
- Use small sample tables

CONCEPTUAL EVALUATION PROBLEM

Example 15: List the number of offerings taught in 2013 by faculty rank and department. Exclude combinations of faculty rank and department with less than two offerings taught.

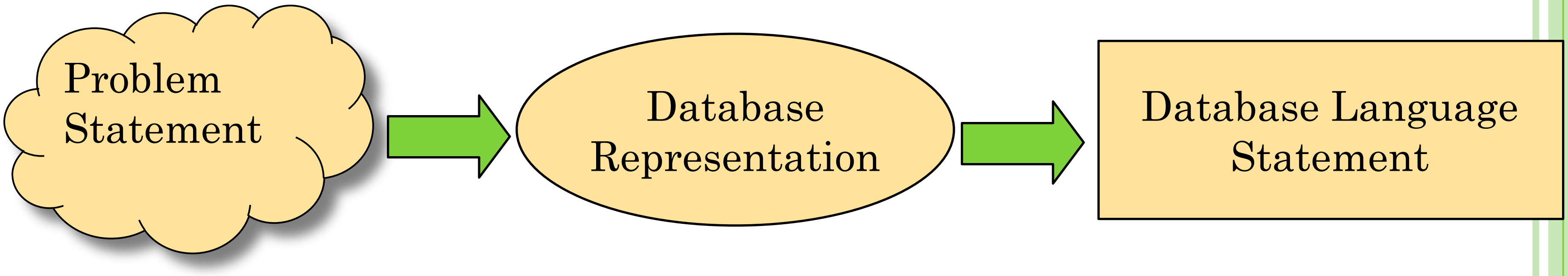
```
SELECT FacRank, FacDept,
       COUNT(*) AS NumOfferings
  FROM Faculty, Offering
  WHERE Offering.FacSSN = Faculty.FacSSN
    AND OffYear = 2013
  GROUP BY FacRank, FacDept
  HAVING COUNT(*) > 1
```

Example 15

```
SELECT FacRank, FacDept,
       COUNT(*) AS NumOfferings
  FROM Faculty, Offering
  WHERE Offering.FacSSN = Faculty.FacSSN
    AND OffYear = 2013
  GROUP BY FacRank, FacDept
  HAVING COUNT(*) > 1
```

FacRank	*FacDept*	*NumOfferings*
Professor	*Computer*	*4*
Lecturer	*Computer*	*6*
Professor	*Accounting*	*3*
Lecturer	*Accounting*	*10*

QUERY FORMULATION PROCESS



CRITICAL QUESTIONS

- What tables?
  - Columns in output
  - Conditions to test (including join conditions)
- How to combine the tables?
  - Usually join PK to FK
  - More complex ways to combine
- Individual rows or groups of rows?
  - Aggregate functions in output
  - Conditions with aggregate functions

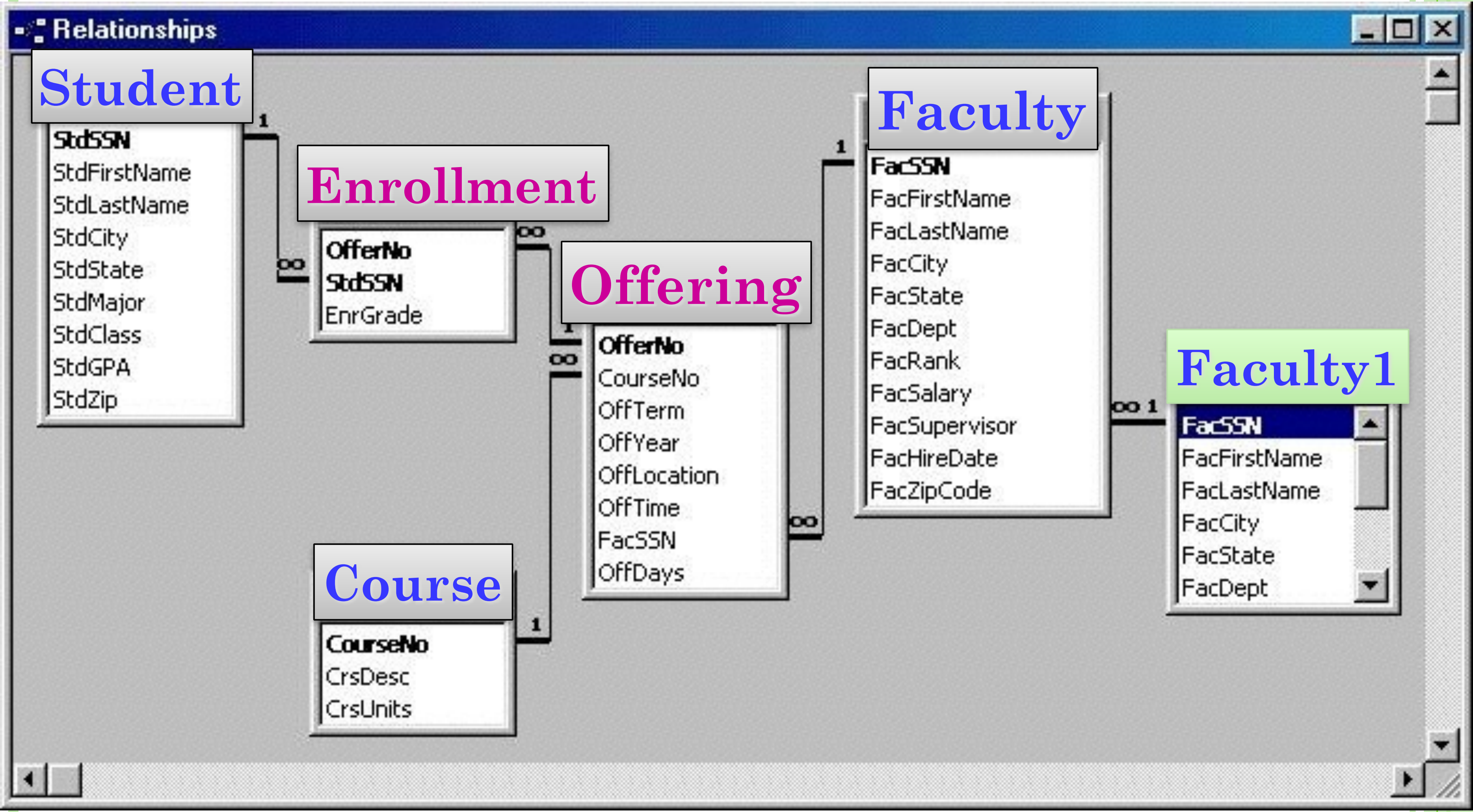
EFFICIENCY CONSIDERATIONS

- Little concern for efficiency
- Intelligent SQL compilers
- Correct and non redundant solution
  - No extra tables
  - No unnecessary grouping
  - Use **HAVING** for group conditions only

ADVANCED PROBLEMS

- Joining multiple tables
- Self joins
- Grouping after joining multiple tables
- Traditional set operators

Example 17: List Professor Vince teaching schedule in fall 2012.



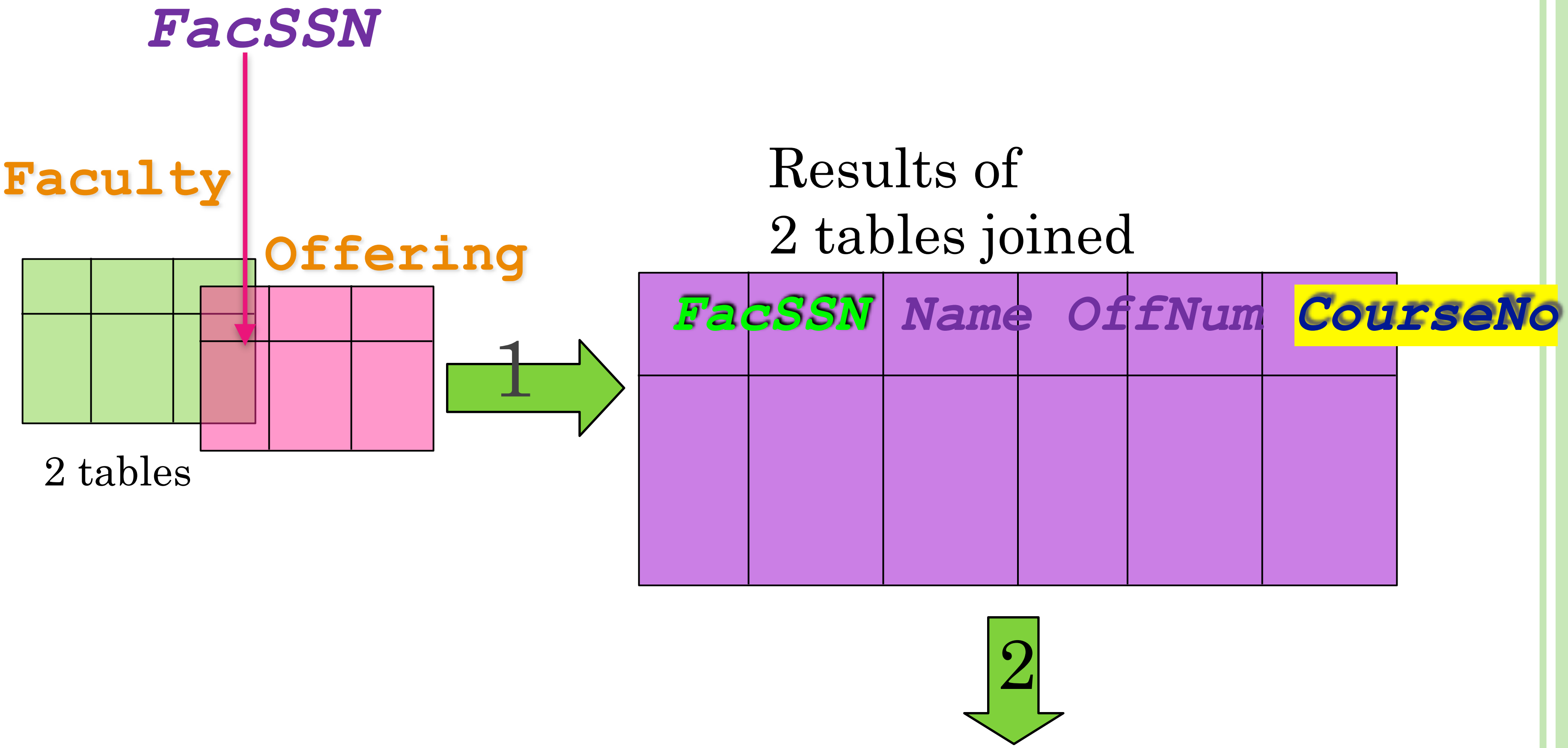
Example 17: List Professor Vince teaching schedule.

For each course, list

- **the offering number,**
- **course number,**
- *number of course units,*
- **days,**
- **location, and**
- **time.**

? Offering X Faculty

Join **Faculty,Offering, Course**



FacSSN

Faculty

Offering

2 tables

1

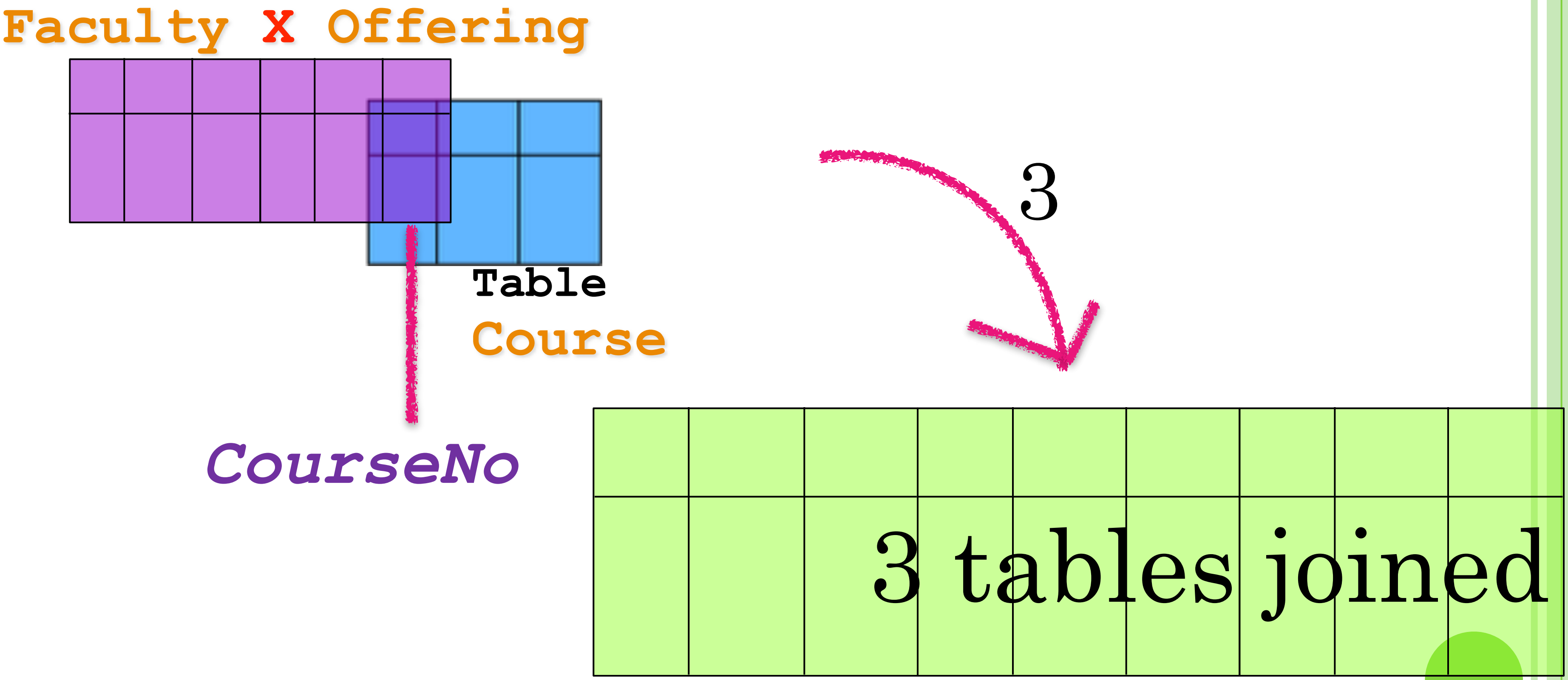
Results of
2 tables joined

FacSSN ***Name*** ***OffNum*** ***CourseNo***

2

Join with **Table** **Course**

Join **Faculty, Offering,Course**



Faculty X Offering

Table

Course

CourseNo

3

3 tables joined

JOINING THREE TABLES

```
SELECT OfferNo, Offering.CourseNo,
OffDays,CrsUnits, OffLocation, OffTime
 FROM Faculty, Offering, Course
 WHERE Faculty.FacSSN=Offering.FacSSN
 AND Offering.CourseNo=Course.CourseNo
```

2nd JOIN

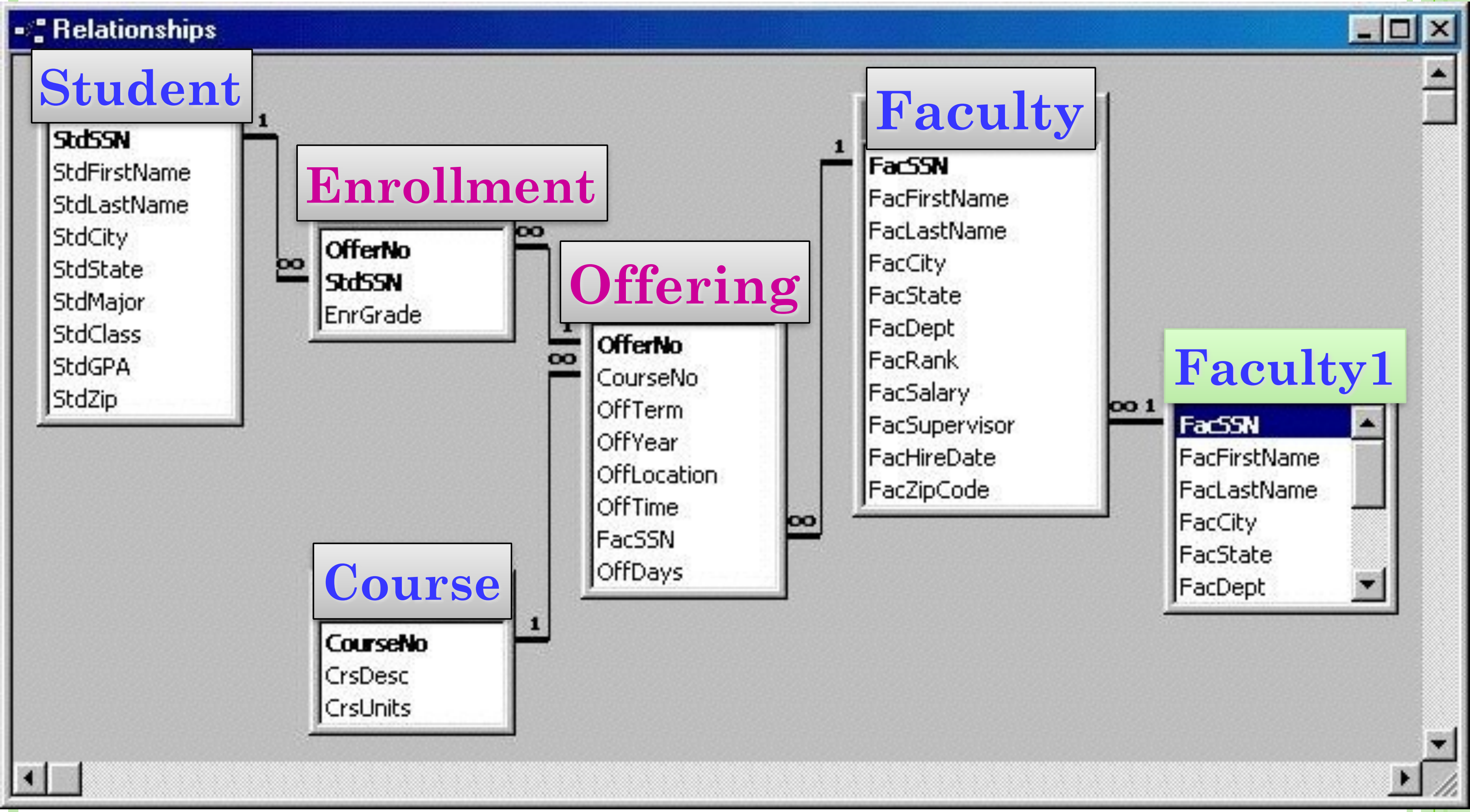
1st JOIN

JOINING THREE TABLES

Example 16: List Leonard Vince's teaching schedule in fall 2012. For each course, list the offering number, course number, number of units, days, location, and time.

```
SELECT OfferNo, Offering.CourseNo, OffDays,
       CrsUnits, OffLocation, OffTime
 FROM Faculty, Offering, Course
 WHERE Faculty.FacSSN = Offering.FacSSN
   AND Offering.CourseNo = Course.CourseNo
   AND OffYear = 2012 AND OffTerm = 'FALL'
   AND FacFirstName = 'Leonard'
   AND FacLastName = 'Vince'
```

Example 17: List Bob Norbert's course schedule in spring 2013.



JOINING FOUR TABLES

Example 17: List Bob Norbert's course schedule in spring 2013.

For each course, list

- the offering number,
- course number,
- days, location, time,
- *faculty name*.

JOINING FOUR TABLES

Example 17: List Bob Norbert's course schedule in spring 2013. For each course, list the offering number, course number, days, location, time, and faculty name.

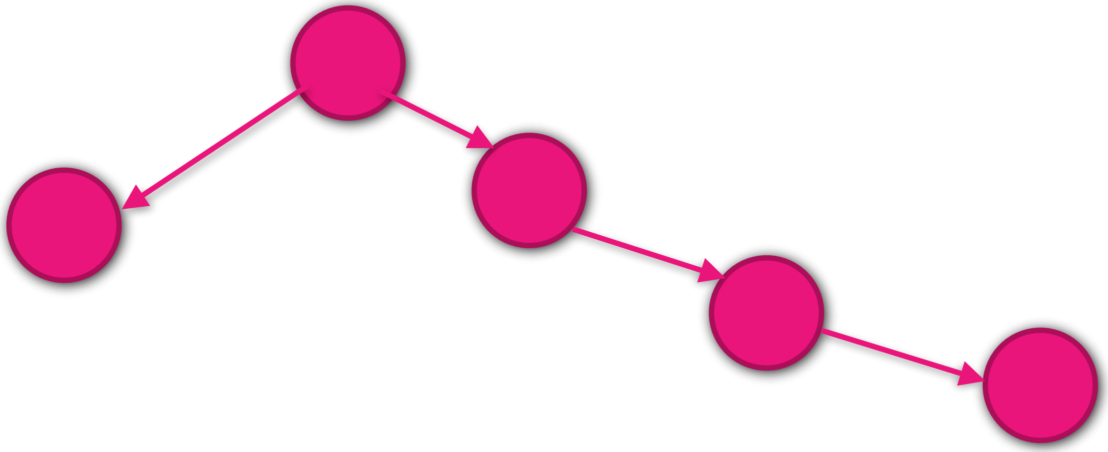
```
SELECT Offering.OfferNo, Offering.CourseNo,
       OffDays, OffLocation, OffTime,
       FacFirstName, FacLastName

 FROM Faculty, Offering, Enrollment, Student

 WHERE Offering.OfferNo = Enrollment.OfferNo
   AND Student.StdSSN = Enrollment.StdSSN
   AND Faculty.FacSSN = Offering.FacSSN
   AND OffYear = 2013 AND OffTerm = 'SPRING'
   AND StdFirstName = 'BOB'
   AND StdLastName = 'NORBERT'
```

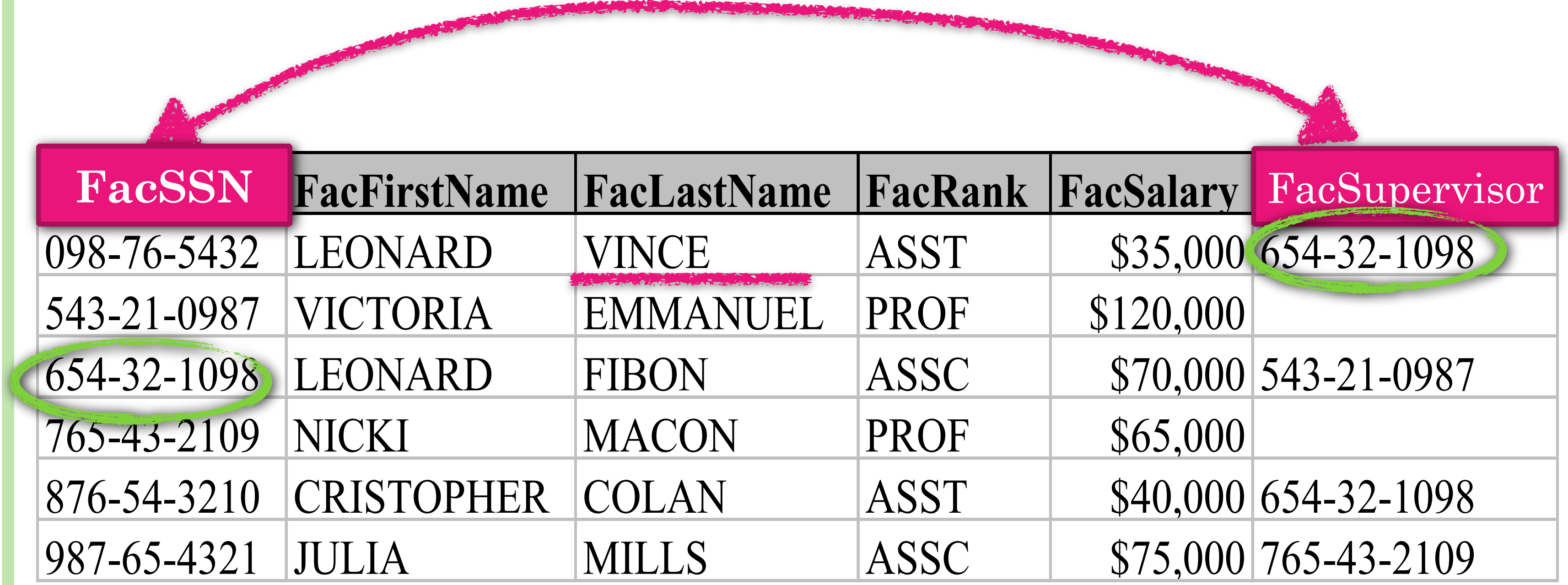
SELF-JOIN

- Join a table to itself
- Usually involve a self-referencing relationship
  - Supervise, Prerequisite-of
- Useful to find relationships among rows of the same table
  - Find subordinates within **a preset** number of levels
  - Find subordinates within **any number** of levels requires embedded SQL



SELF-JOIN EXAMPLE

Example 18.1: List faculty members and their supervisor.



FacSSN	FacFirstName	FacLastName	FacRank	FacSalary	FacSupervisor
098-76-5432	LEONARD	VINCE	ASST	$35,000	654-32-1098
543-21-0987	VICTORIA	EMMANUEL	PROF	$120,000	
654-32-1098	LEONARD	FIBON	ASSC	$70,000	543-21-0987
765-43-2109	NICKI	MACON	PROF	$65,000	
876-54-3210	CRISTOPHER	COLAN	ASST	$40,000	654-32-1098
987-65-4321	JULIA	MILLS	ASSC	$75,000	765-43-2109

Example 18.1 Join Faculty1 with Faculty2

FK

Faculty1

FacSSN	FacFirstName	FacLastName	FacRank	FacSalary	FacSupervisor
098-76-5432	LEONARD	VINCE	ASST	$35,000	654-32-1098
543-21-0987	VICTORIA	EMMANUEL	PROF	$120,000	
654-32-1098	LEONARD	FIBON	ASSC	$70,000	543-21-0987
765-43-2109	NICKI	MACON	PROF	$65,000	
876-54-3210	CRISTOPHER	COLAN	ASST	$40,000	654-32-1098
987-65-4321	JULIA	MILLS	ASSC	$75,000	765-43-2109

PK

Faculty2

FacSSN	FacFirstName	FacLastName	FacRank	FacSalary	FacSupervisor
098-76-5432	LEONARD	VINCE	ASST	$35,000	654-32-1098
543-21-0987	VICTORIA	EMMANUEL	PROF	$120,000	
654-32-1098	LEONARD	FIBON	ASSC	$70,000	543-21-0987
765-43-2109	NICKI	MACON	PROF	$65,000	
876-54-3210	CRISTOPHER	COLAN	ASST	$40,000	654-32-1098
987-65-4321	JULIA	MILLS	ASSC	$75,000	765-43-2109

Faculty1.*FacSupervisor* = Faculty2.*FacSSN*

Example 18.1: SQL

FacSSN	FacFirstName	FacLastName	FacRank	FacSalary	FacSupervisor	SupLname
098-76-5432	LEONARD	VINCE	ASST	$35,000	654-32-1098	Fibon
543-21-0987	VICTORIA	EMMANUEL	PROF	$120,000		
654-32-1098	LEONARD	FIBON	ASSC	$70,000	543-21-0987	Emmanuel
765-43-2109	NICKI	MACON	PROF	$65,000		
876-54-3210	CRISTOPHER	COLAN	ASST	$40,000	654-32-1098	Fibon
987-65-4321	JULIA	MILLS	ASSC	$75,000	765-43-2109	Macon

```
SELECT f.FacSSN,f.FacLastName,
       f.FacSalary,f.FacSupervisor,
S.FacLastName AS SupLname
 FROM Faculty f, Faculty S
 WHERE f.FacSupervisor = S.FacSSN
```

SELF-JOIN EXAMPLE

Example 18.2: List faculty members who have a **higher salary** than their supervisor.

FacSSN	FacFirstName	FacLastName	FacRank	FacSalary	FacSupervisor	Salary
098-76-5432	LEONARD	VINCE	ASST	$35,000	654-32-1098	70000
543-21-0987	VICTORIA	EMMANUEL	PROF	$120,000		
654-32-1098	LEONARD	FIBON	ASSC	$70,000	543-21-0987	120000
765-43-2109	NICKI	MACON	PROF	$65,000		
876-54-3210	CRISTOPHER	COLAN	ASST	$40,000	654-32-1098	70000
987-65-4321	JULIA	MILLS	ASSC	$75,000	765-43-2109	65000

SELF-JOIN EXAMPLE

Example 18:

```
SELECT f.FacSSN, f.FacLastName,
       f.FacSalary, S.FacSSN,
       f2.FacLastName, S.FacSalary
 FROM Faculty f, Faculty S
 WHERE f.FacSupervisor = S.FacSSN
   AND f.FacSalary > S.FacSalary
```

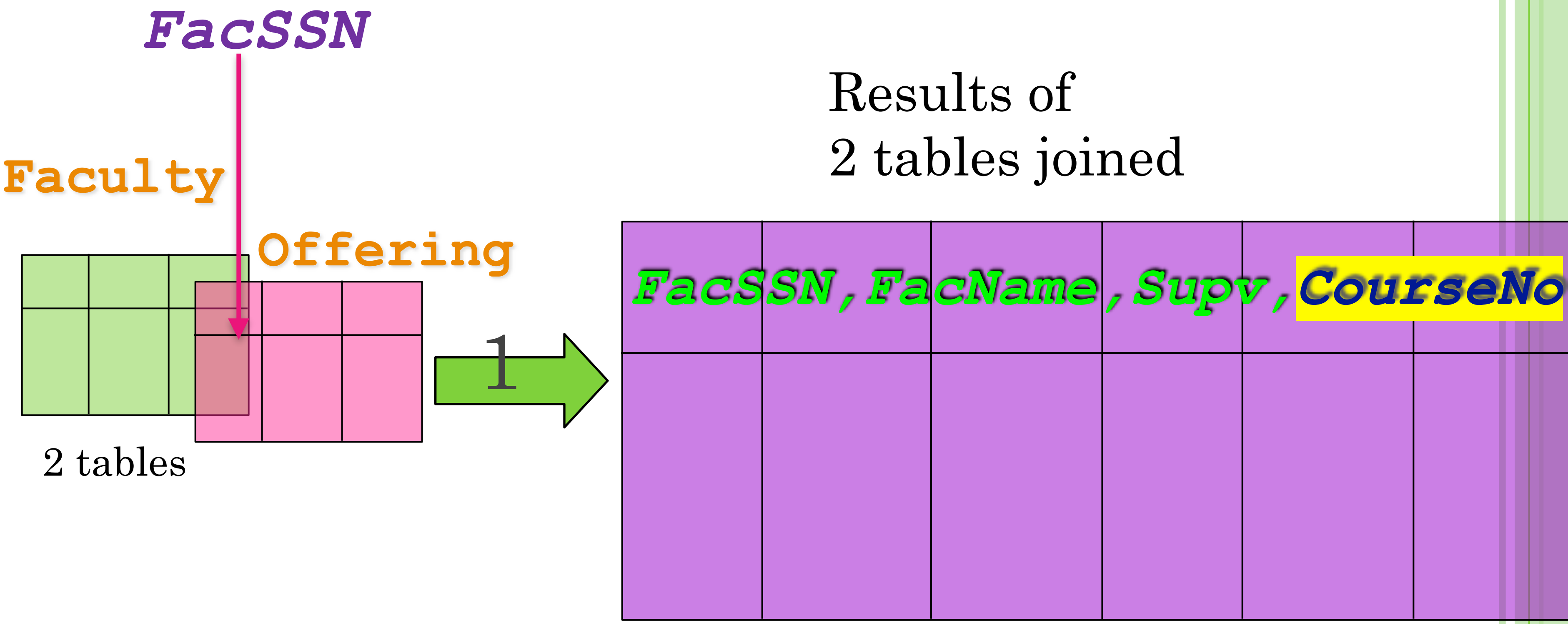
JOIN

MULTIPLE JOINS BETWEEN TABLES

Example 19: List the names of faculty members and the course number for which the faculty member teaches the same course number as his or her supervisor

- **Faculty**
- **Offering**

Join **Faculty,Offering**



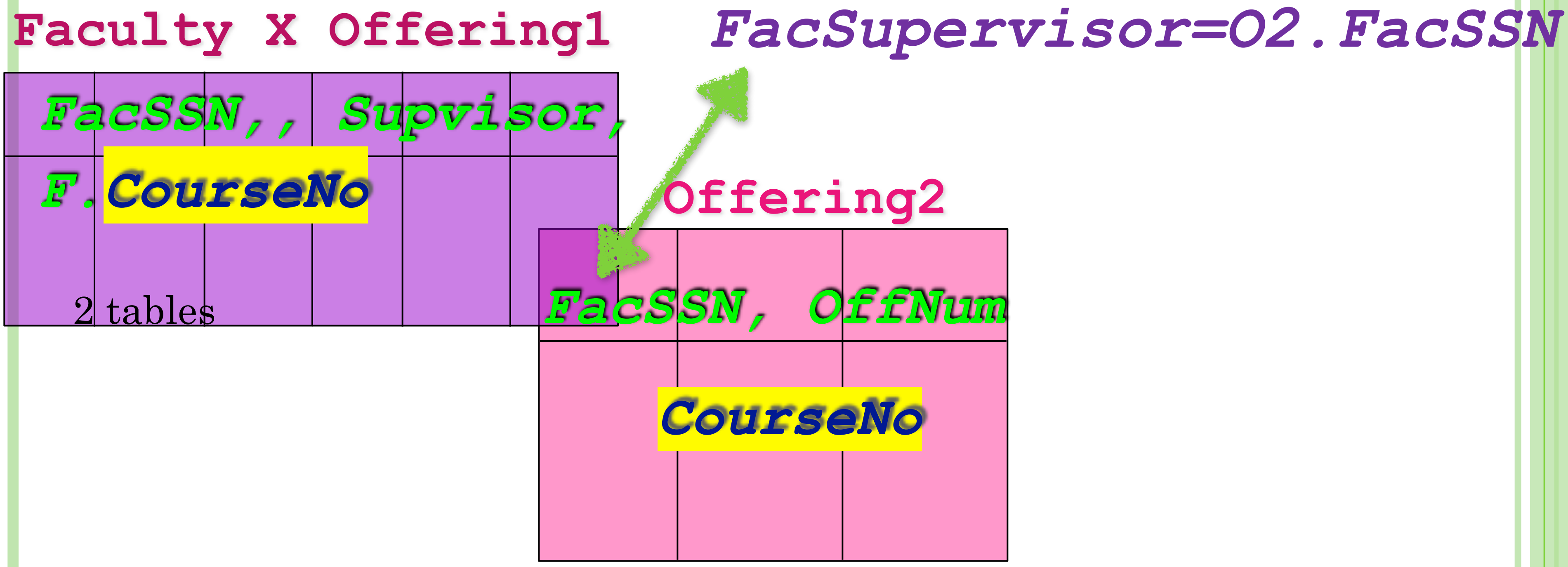
2 tables

Results of
2 tables joined

FacSSN,FacName,Supv,CourseNo

What about courses taught by their supervisor?

Join **Faculty,Offering1,Offering2**



Faculty X Offering1

FacSupervisor=O2.FacSSN

FacSSN,, Supvisor,

F.CourseNo

2 tables

Offering2

FacSSN, OffNum

CourseNo

Courses taught by their supervisor!

MULTIPLE JOINS BETWEEN TABLES

Join Faculty and Offering O1

```
WHERE Faculty.FacSSN = O1.FacSSN
```

FacSSN	SupervisorSSN	Course	
1	3	**351**	
1	3	**111**	
2	3	101	

course that Faculties teach

Faculty member and supervisor teach the same course number

```
JOIN Faculty, Offering O1, Offering O2
 WHERE Faculty.FacSSN = O1.FacSSN
   AND Faculty.FacSupervisor = O2.FacSSN
```

Courses taught by their supervisor!

FacSSN	SupervisorSSN	O1.Course	O2.Course	
1	3	351	111	
1	3	351	501	
1	3	111	111	
1	3	111	501	
2	3	101	111	
2	3	101	501	

MULTIPLE JOINS BETWEEN TABLES

Example 19:

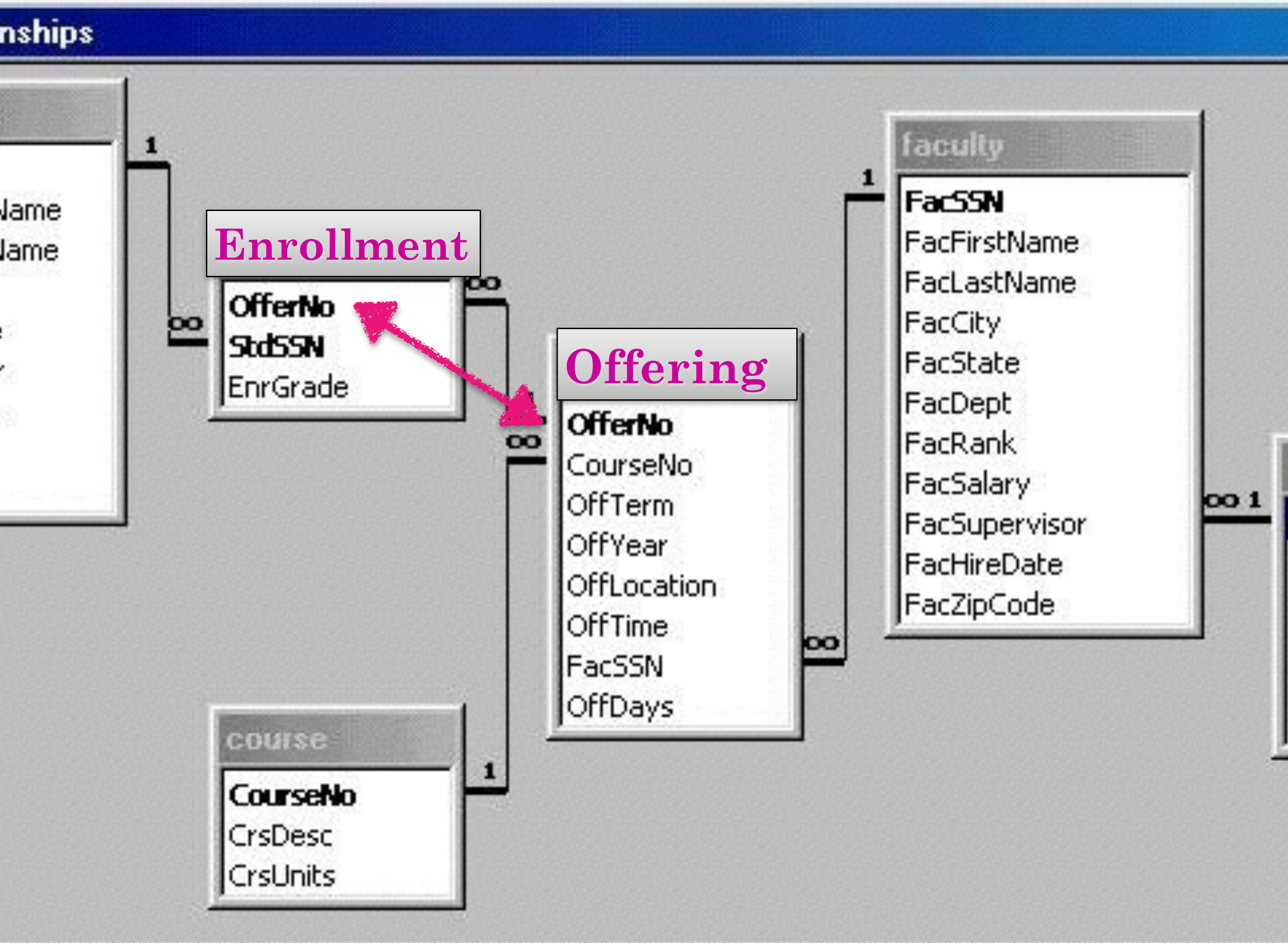
```
SELECT FacFirstName, FacLastName, O1.CourseNo
 FROM Faculty, Offering O1, Offering O2
 WHERE Faculty.FacSSN = O1.FacSSN
   AND Faculty.FacSupervisor = O2.FacSSN
   AND O1.OffYear = 2013 AND O2.OffYear=2013
   AND O1.CourseNo = O2.CourseNo
```

MULTIPLE COLUMN GROUPING

Example 20: List the course number, the offering number, and the number of students enrolled. Only include courses offered in spring 2013.

```
SELECT CourseNo, Enrollment.OfferNo,
       Count(*) AS NumStudents
 FROM Offering, Enrollment
 WHERE Offering.OfferNo = Enrollment.OfferNo
   AND OffYear = 2013 AND OffTerm = 'SPRING'
 GROUP BY Enrollment.OfferNo, CourseNo
```

Note: any columns appearing in SELECT must be either a grouping column or an aggregate expression



Enrollment

Offering

Join Enrollment E and Offering O

WHERE E.OfferNo = O.OfferNo

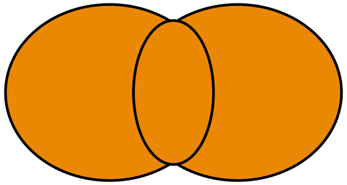
E.OfferNo	E.StdSSN	O.CourseNo	
1	5432	**351**	1
1	6789	**351**	
2	1234	111	2
3	5432	499	3
3	1234	499	

MULTIP COLUMN GROUPING

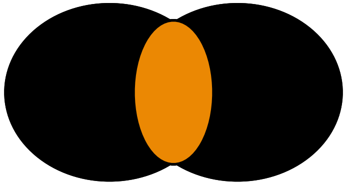
```
SELECT CourseNo, Enrollment.OfferNo,
       Count(*) AS NumStudents
 FROM Offering, Enrollment
 WHERE Offering.OfferNo = Enrollment.OfferNo
   AND OffYear = 2013 AND OffTerm = 'SPRING'
 GROUP BY Enrollment.OfferNo, CourseNo
```

Note: any columns appearing in SELECT must be either a grouping column or an aggregate expression

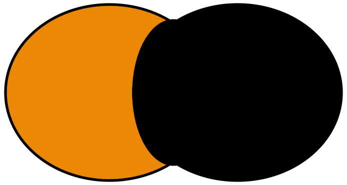
TRADITIONAL SET OPERATORS



A UNION B



A INTERSECT B



A MINUS B

UNION COMPATIBILITY

A	B	C
1	3	**DB**
1	4	**IE**
2	3	ACCT

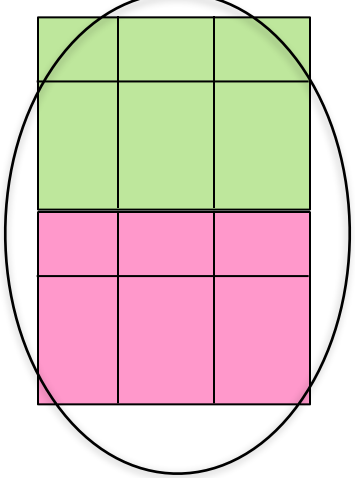
same numbers, names and types compatible

A	B	C
2	3	**ACCT**
9	7	**IS**
10	7	EE

A	B	C
1	3	**DB**
1	4	**IE**
2	3	ACCT

UNION

A	B	C
2	3	**ACCT**
9	7	**IS**
10	7	EE

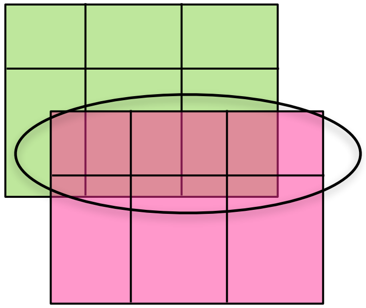


A	B	C
1	3	**DB**
1	4	**IE**
2	3	ACCT
9	7	**IS**
10	7	EE

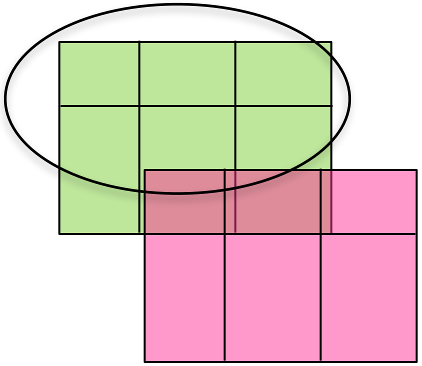
A	B	C
1	3	**DB**
1	3	**DB**
2	3	ACCT

Intersect

A	B	C
2	3	**ACCT**
9	7	**IS**
10	7	EE



A	B	C
2	3	ACCT



A	B	C
1	3	DB
1	4	IE
2	3	ACCT

Minus

A	B	C
2	3	ACCT
9	7	IS
10	7	EE

A	B	C
1	3	DB
1	4	IE

Set Operator COMPATIBILITY

- Requirement for the traditional set operators
- Strong requirement
  - Same number of columns
  - Each corresponding column is compatible
  - Positional correspondence
- Apply to similar tables by removing columns first

SQL UNION EXAMPLE

Example 21: Retrieve basic data about all university people

```
SELECT FacSSN AS SSN, FacFirstName AS FirstName,
       FacLastName AS LastName, FacCity AS City,
       FacState AS State
 FROM Faculty
     UNION
SELECT StdSSN AS SSN, StdFirstName AS FirstName,
       StdLastName AS LastName, StdCity AS City,
       StdState AS State
 FROM Student
```

ORACLE INTERSECT EXAMPLE

Example 22: Show teaching assistants, faculty who are students. Only show the common columns in the result.

```
SELECT FacSSN AS SSN, FacFirstName AS
       FirstName, FacLastName AS LastName,
       FacCity AS City, FacState AS State
 FROM Faculty
   INTERSECT
SELECT StdSSN AS SSN, StdFirstName AS
       FirstName, StdLastName AS LastName,
       StdCity AS City, StdState AS State
 FROM Student
```

ORACLE MINUS EXAMPLE

Example 23: Show faculty who are not students (pure faculty). Only show the common columns in the result.

```
SELECT FacSSN AS SSN, FacFirstName AS
       FirstName, FacLastName AS LastName,
       FacCity AS City, FacState AS State
 FROM Faculty
   MINUS
SELECT StdSSN AS SSN, StdFirstName AS
       FirstName, StdLastName AS LastName,
       StdCity AS City, StdState AS State
 FROM Student
```

DATA MANIPULATION STATEMENTS

- INSERT: adds one or more rows
- UPDATE: modifies one or more rows
- DELETE: removes one or more rows
- Use SELECT statement to INSERT multiple rows
- UPDATE and DELETE can use a WHERE clause
- **Not as widely used as SELECT statement**

INSERT EXAMPLE

Example 24: Insert a row into the *Student* table supplying values for all columns.

```
INSERT INTO Student
  (StdSSN, StdFirstName, StdLastName,
   StdCity, StdState, StdZip, StdClass,
StdMajor, StdGPA)
 VALUES
('999999999','JOE','STUDENT','SEATAC',
   'WA','98042-1121','FR','IS', 0.0)
```

UPDATE EXAMPLE

Example 25: Change the major and class of Homer Wells.

```
UPDATE Student
 SET StdMajor = 'ACCT',
     StdClass = 'SO'
 WHERE StdFirstName = 'HOMER'
   AND StdLastName = 'WELLS'
```

INSERT EXAMPLE

```
INSERT INTO Student
  (StdSSN, StdFirstName, StdLastName,
   StdCity, StdState, StdZip, StdClass,
StdMajor, StdGPA)
 SELECT <matched columns>
   FROM …
   WHERE ….
     ……. .
```

DELETE EXAMPLE

Example 26: Delete all IS majors who are seniors.

```
DELETE FROM Student
 WHERE StdMajor = 'IS'
   AND StdClass = 'SR'
```

SUMMARY

- SQL is a broad language
- SELECT statement is complex
- Use problem solving guidelines
- **Lots of practice** to master query formulation and SQL