Chapter 3

The Relational Data Model

Outline

- Relational model basics
- Integrity rules
- Rules about referenced rows
- Relational Algebra

Tables

- Relational database is a collection of tables
- Heading: table name and column names
- Body: rows, occurrences of data

Student

StdSSN	StdLastName	StdMajor	StdClass	StdGPA
123-45-6789	WELLS	IS	FR	3.00
124-56-7890	NORBERT	FIN	JR	2.70
234-56-7890	KENDALL	ACCT	JR	3.50

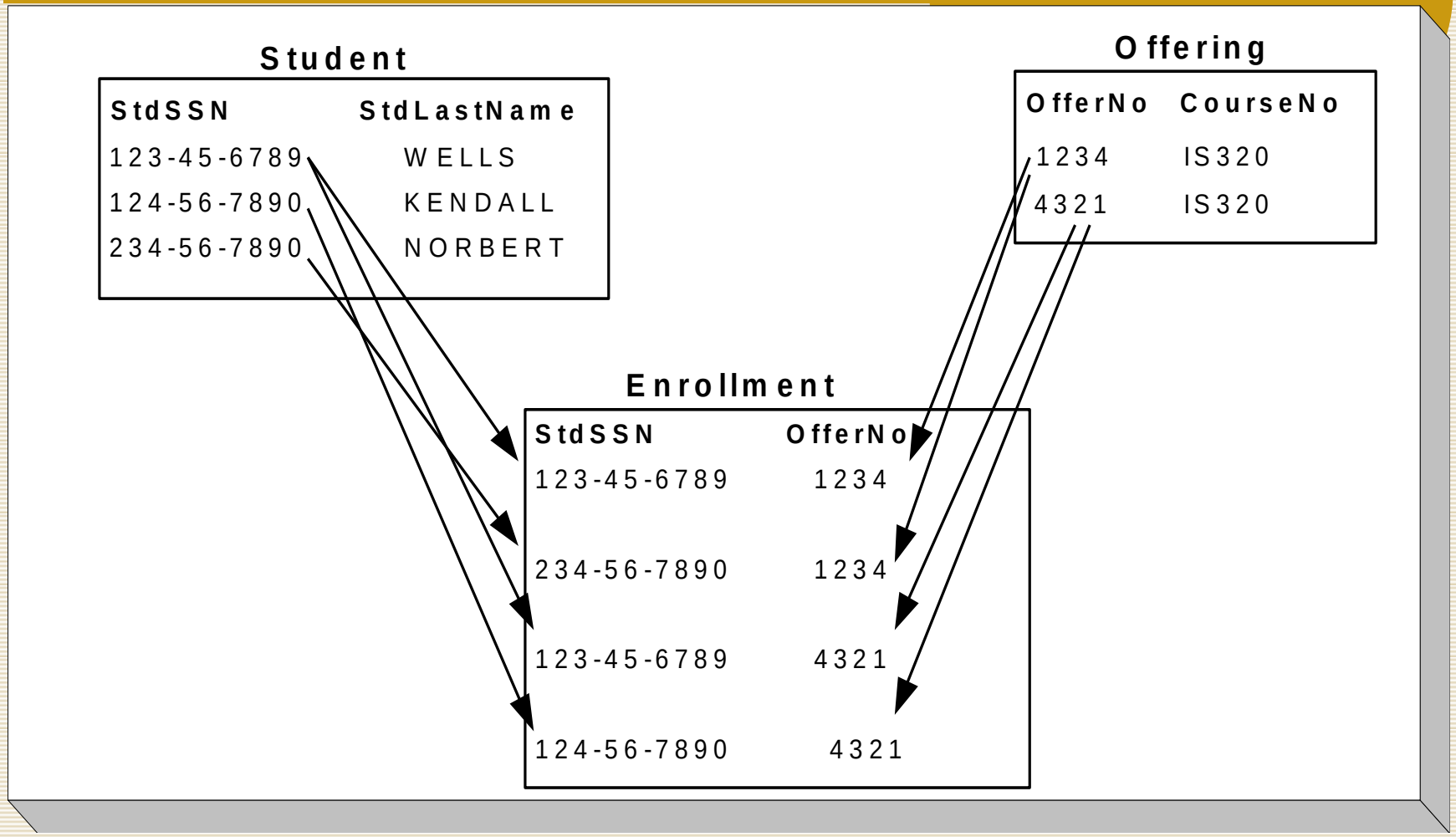
CREATE TABLE Statement

```
CREATE TABLE Student
(     StdSSN            CHAR(11),
      StdFirstName      VARCHAR(50),
      StdLastName       VARCHAR(50),
      StdCity           VARCHAR(50),
      StdState          CHAR(2),
      StdZip            CHAR(10),
      StdMajor          CHAR(6),
      StdClass          CHAR(6),
      StdGPA            DECIMAL(3,2)    )
```

Common Data Types

- CHAR(L)
- VARCHAR(L)
- INTEGER
- FLOAT(P)
- Date/Time: DATE, TIME, TIMESTAMP
- DECIMAL(W, R)
- BOOLEAN

Relationships



Alternative Terminology

Table-oriented	**Set-oriented**	**Record-oriented**
Table	Relation	Record-type, file
Row	Tuple	Record
Column	Attribute	Field

Integrity Rules

- Entity integrity: primary keys
  - Each table has column(s) with unique values
  - Ensures entities are traceable
- Referential integrity: foreign keys
  - Values of a column in one table match values in a source table
  - Ensures valid references among tables

Formal Definitions I

- **Superkey**: column(s) with unique values
- **Candidate key**: minimal superkey
- **Null value**: special value meaning value unknown or inapplicable
- **Primary key**: a designated candidate key; cannot contain null values
- **Foreign key**: column(s) whose values must match the values in a candidate key of another table

Formal Definitions II

- **Entity integrity**
  - No two rows with the same primary key value
  - No null values in any part of a primary key
- **Referential integrity**
  - Foreign keys must match candidate key of source table
  - Foreign keys can be null in some cases
  - In SQL, foreign keys associated with primary keys

Course Table Example

```
CREATE TABLE Course
(     CourseNo          CHAR(6),
      CrsDesc           VARCHAR(250),
      CrsUnits          SMALLINT,
 CONSTRAINT PKCourse PRIMARY KEY(CourseNo),
 CONSTRAINT UniqueCrsDesc UNIQUE (CrsDesc) )
```

Enrollment Table Example

```
CREATE TABLE Enrollment
(     OfferNo    INTEGER,
      StdSSN     CHAR(11),
      EnrGrade   DECIMAL(3,2),
CONSTRAINT PKEnrollment PRIMARY KEY
   (OfferNo, StdSSN),
CONSTRAINT FKOfferNo FOREIGN KEY (OfferNo)
    REFERENCES Offering,
CONSTRAINT FKStdSSN FOREIGN KEY (StdSSN)
   REFERENCES Student )
```

Offering Table Example

```
CREATE TABLE Offering
( OfferNo      INTEGER,
  CourseNo     CHAR(6) CONSTRAINT OffCourseNoRequired NOT
   NULL,
  OffLocation VARCHAR(50),
  OffDays      CHAR(6),
  OffTerm      CHAR(6) CONSTRAINT OffTermRequired
      NOT NULL,
   OffYear     INTEGER CONSTRAINT OffYearRequired
      NOT NULL,
  FacSSN       CHAR(11),
   OffTime     DATE,
CONSTRAINT PKOffering PRIMARY KEY (OfferNo),
CONSTRAINT FKCourseNo FOREIGN KEY (CourseNo)
   REFERENCES Course,
CONSTRAINT FKFacSSN FOREIGN KEY (FacSSN)
   REFERENCES Faculty )
```

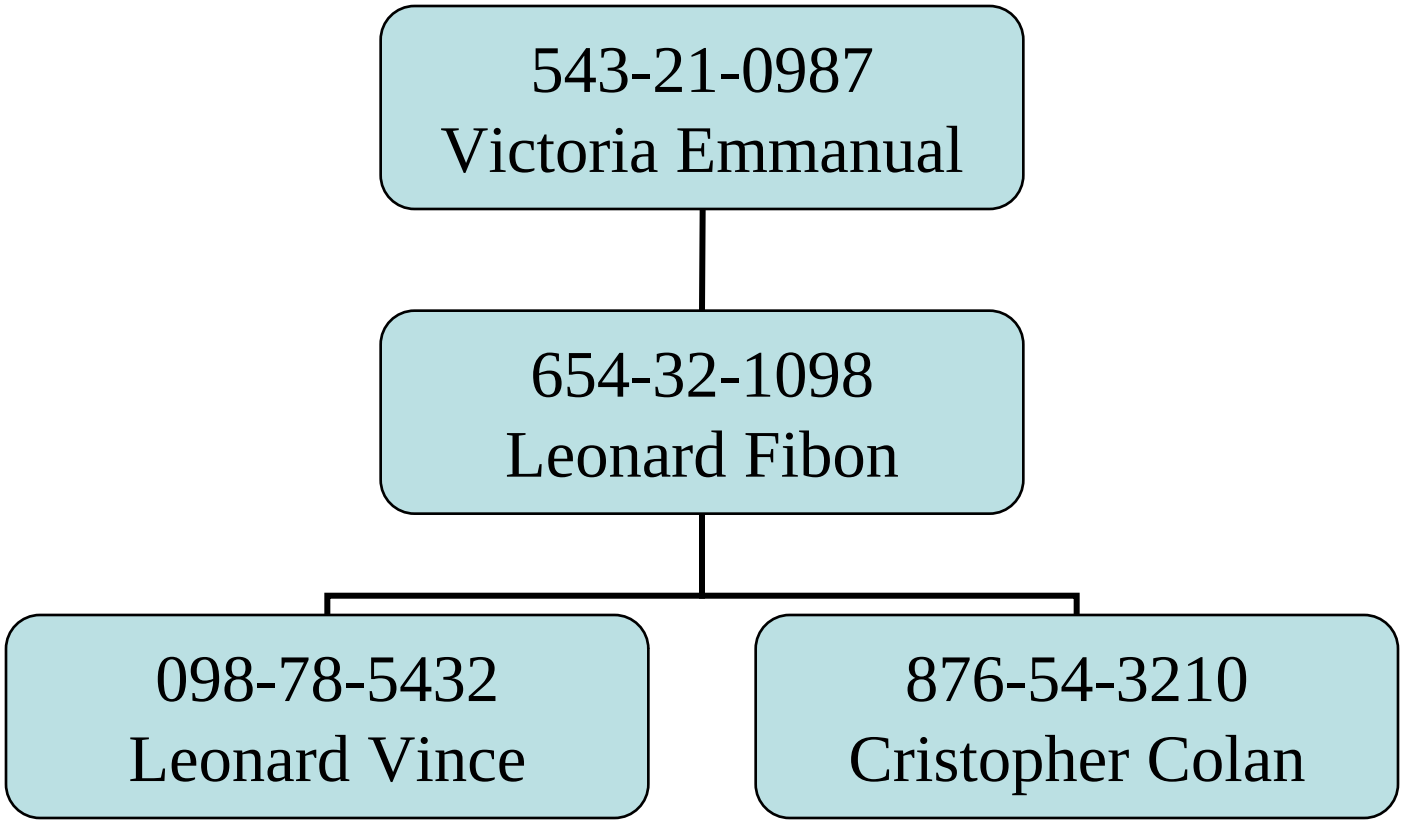
Self-Referencing Relationships

- Foreign key that references the same table
- Represents relationships among members of the same set
- Not common but important in specialized situations

Faculty Data

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

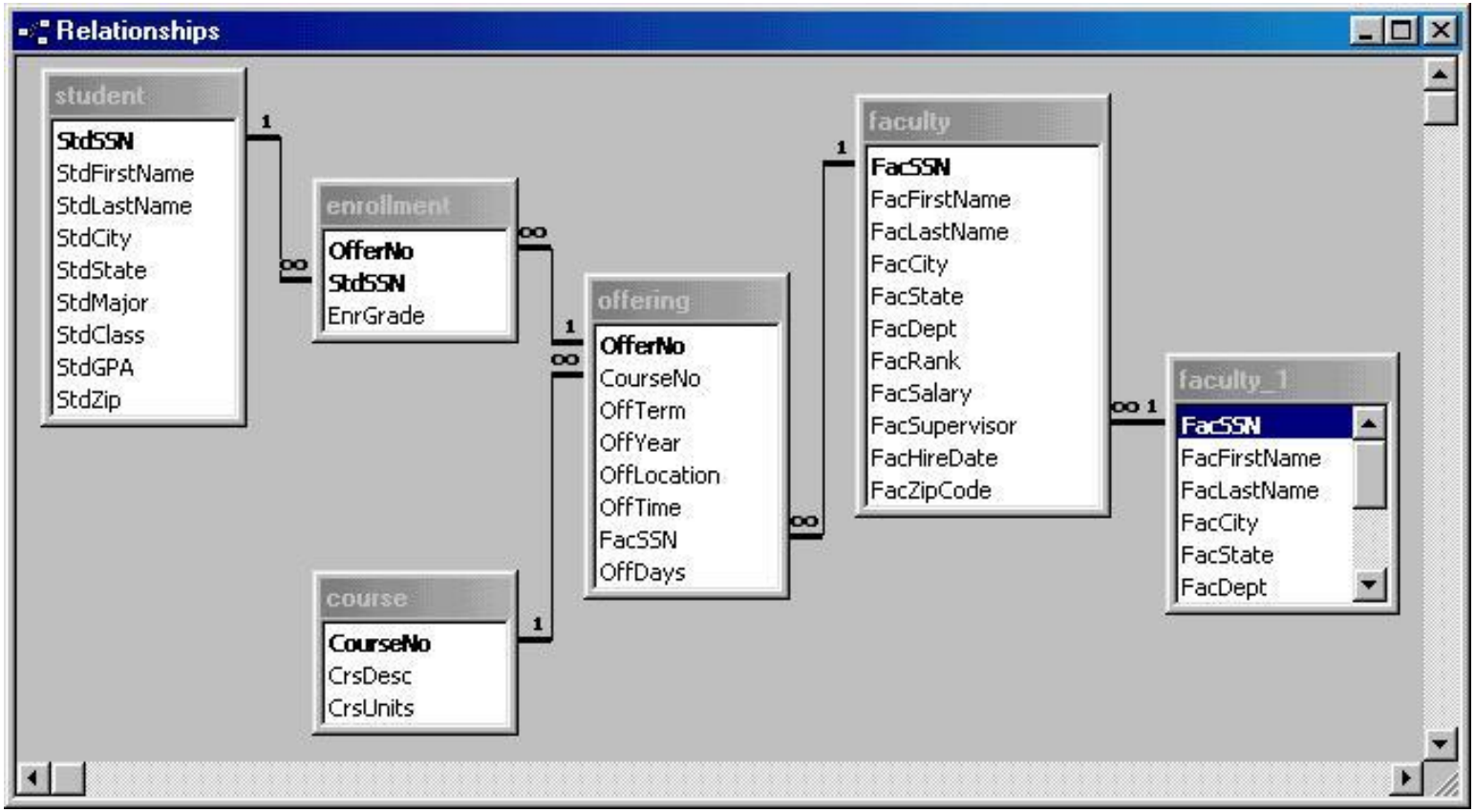
Hierarchical Data Display



Faculty Table Definition

```
CREATE TABLE Faculty
(     FacSSN            CHAR(11),
      FacFirstName      VARCHAR(50) NOT NULL,
      FacLastName       VARCHAR(50) NOT NULL,
      FacCity           VARCHAR(50) NOT NULL,
      FacState          CHAR(2) NOT NULL,
      FacZipCode        CHAR(10)NOT NULL,
      FacHireDate       DATE,
      FacDept           CHAR(6),
      FacSupervisor     CHAR(11),
CONSTRAINT PKFaculty PRIMARY KEY (FacSSN),
CONSTRAINT FKFacSupervisor FOREIGN KEY
   (FacSupervisor) REFERENCES Faculty    )
```

Relationship Window with 1-M Relationships



M-N Relationships

- Rows of each table are related to multiple rows of the other table
- Not directly represented in the relational model
- Use two 1-M relationships and an associative table

Referenced Rows

- Referenced row
  - Foreign keys reference rows in the associated primary key table
  - Enrollment rows refer to Student and Offering
- Actions on referenced rows
  - Delete a referenced row
  - Change the primary key of a referenced row
  - Referential integrity should not be violated

Possible Actions

- Restrict: do not permit action on the referenced row
- Cascade: perform action on related rows
- Nullify: only valid if foreign keys accept null values
- Default: set foreign keys to a default value

SQL Syntax for Actions

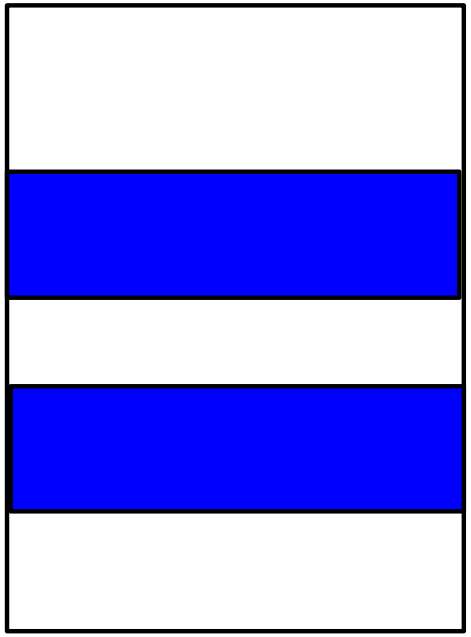
```
CREATE TABLE Enrollment
( OfferNo   INTEGER  NOT NULL,
  StdSSN    CHAR(11) NOT NULL,
      EnrGrade    DECIMAL(3,2),
CONSTRAINT PKEnrollment PRIMARY KEY(OfferNo,
  StdSSN),
CONSTRAINT FKOfferNo FOREIGN KEY (OfferNo)
  REFERENCES Offering
  ON DELETE RESTRICT
  ON UPDATE CASCADE,
CONSTRAINT FKStdSSN FOREIGN KEY (StdSSN) REFERENCES
  Student
  ON DELETE RESTRICT
  ON UPDATE CASCADE )
```

Relational Algebra Overview

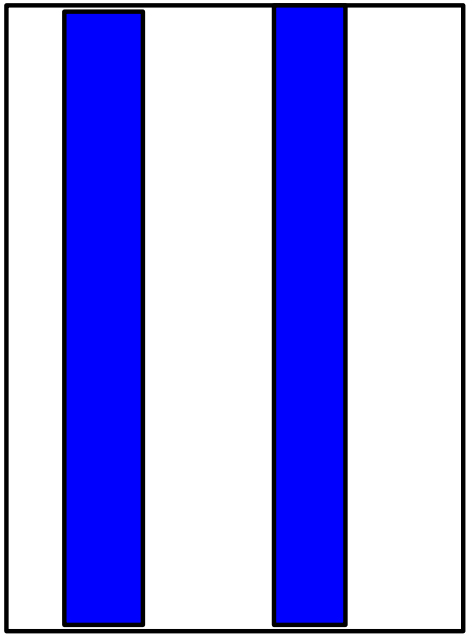
- Collection of table operators
- Transform one or two tables into a new table
- Understand operators in isolation
- Classification
  - Table specific operators
  - Traditional set operators
  - Advanced operators

Subset Operators

Restrict



Project



Subset Operator Notes

- Restrict
  - Logical expression as input
  - Example: OffDays = 'MW' AND OffTerm = 'SPRING' AND OffYear = 2006
- Project
  - List of columns is input
  - Duplicate rows eliminated if present
- Often used together

Extended Cross Product

- Building block for join operator
- Builds a table consisting of all combinations of rows from each of the two input tables
- Produces excessive data
- Subset of cross product is useful (join)

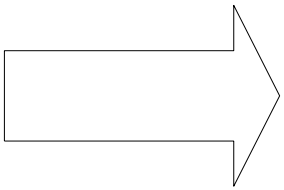
Extended Cross Product Example

Faculty

FacSSN
111-11-1111
222-22-2222
333-33-3333

Student

StdSSN
111-11-1111
444-44-4444
555-55-5555



Faculty PRODUCT Student

FacSSN	StdSSN
111-11-1111	111-11-1111
111-11-1111	444-44-4444
111-11-1111	555-55-5555
222-22-2222	111-11-1111
222-22-2222	444-44-4444
222-22-2222	555-55-5555
333-33-3333	111-11-1111
333-33-3333	444-44-4444
333-33-3333	555-55-5555

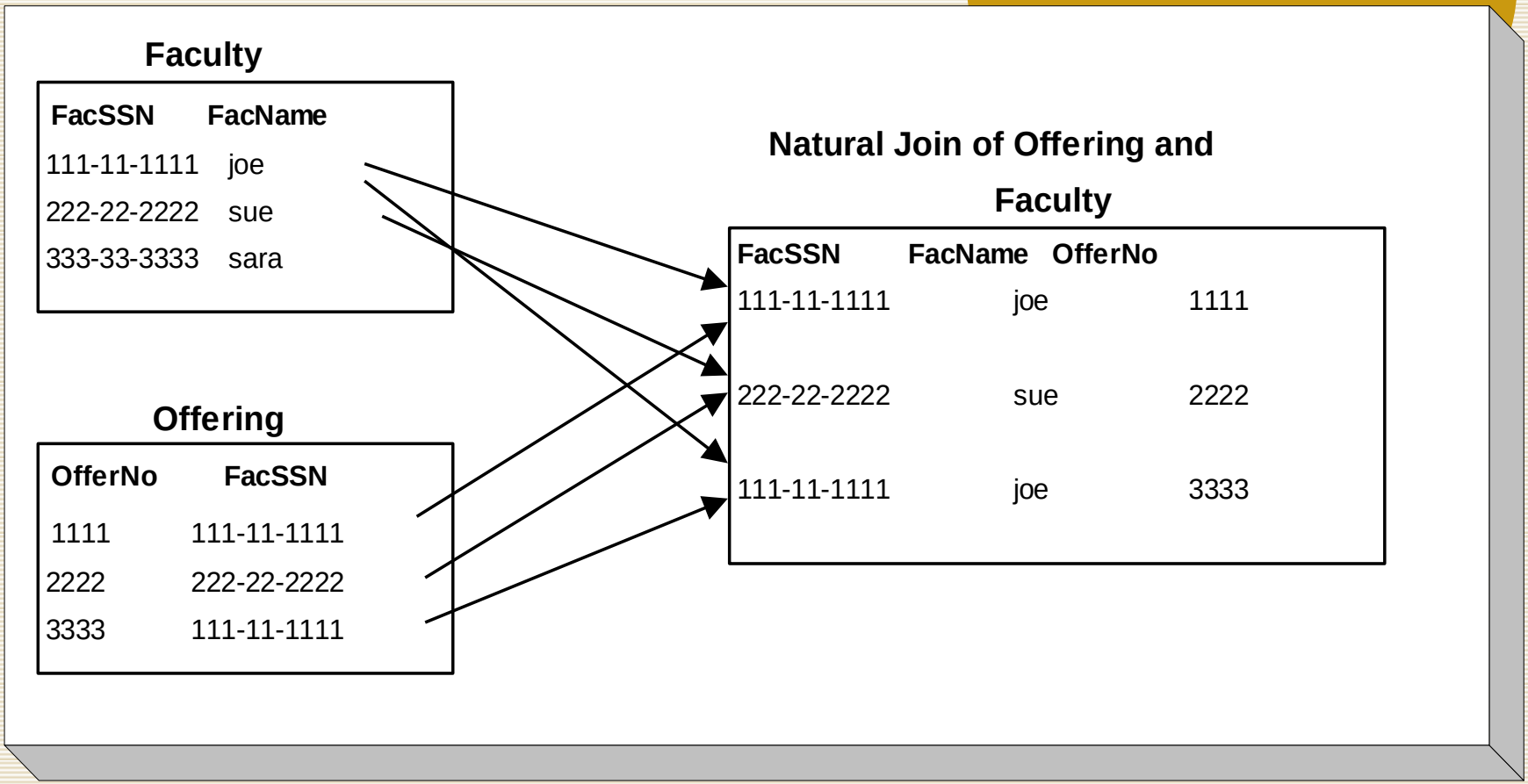
Join Operator

- 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

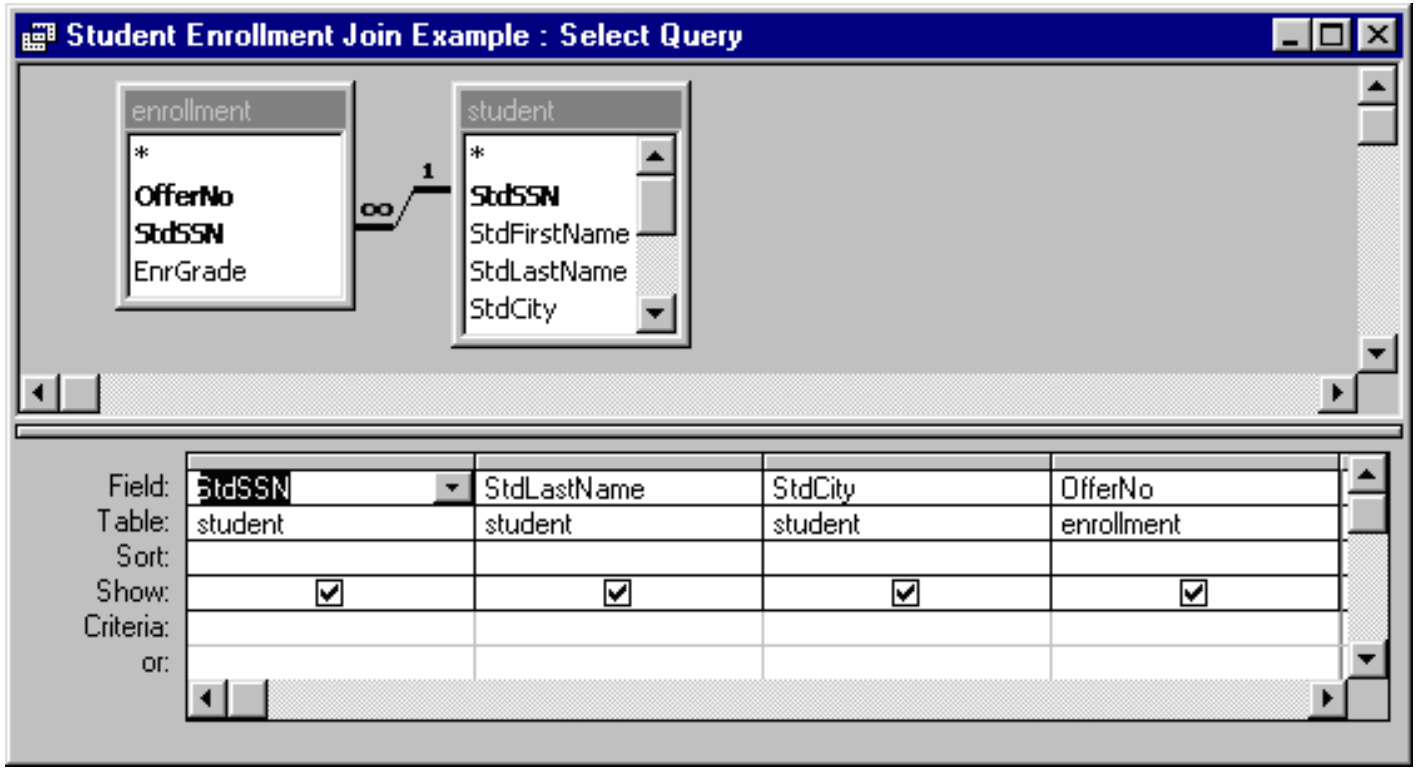
Natural Join Operator

- Most common join operator
- Requirements
  - Equality matching condition
  - Matching columns with the same unqualified names
  - Remove one join column in the result
- Usually performed on PK-FK join columns

Natural Join Example



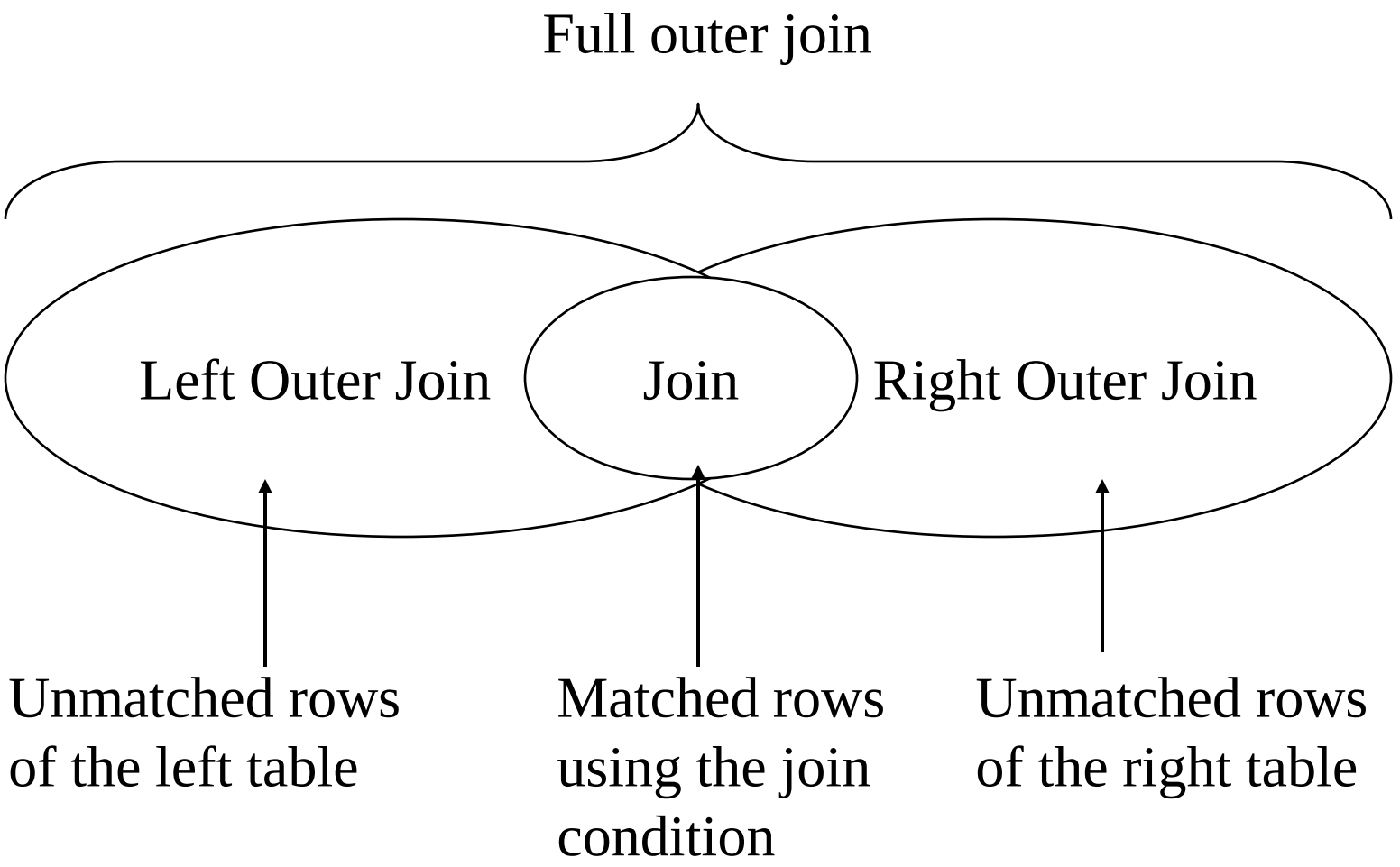
Visual Formulation of Join



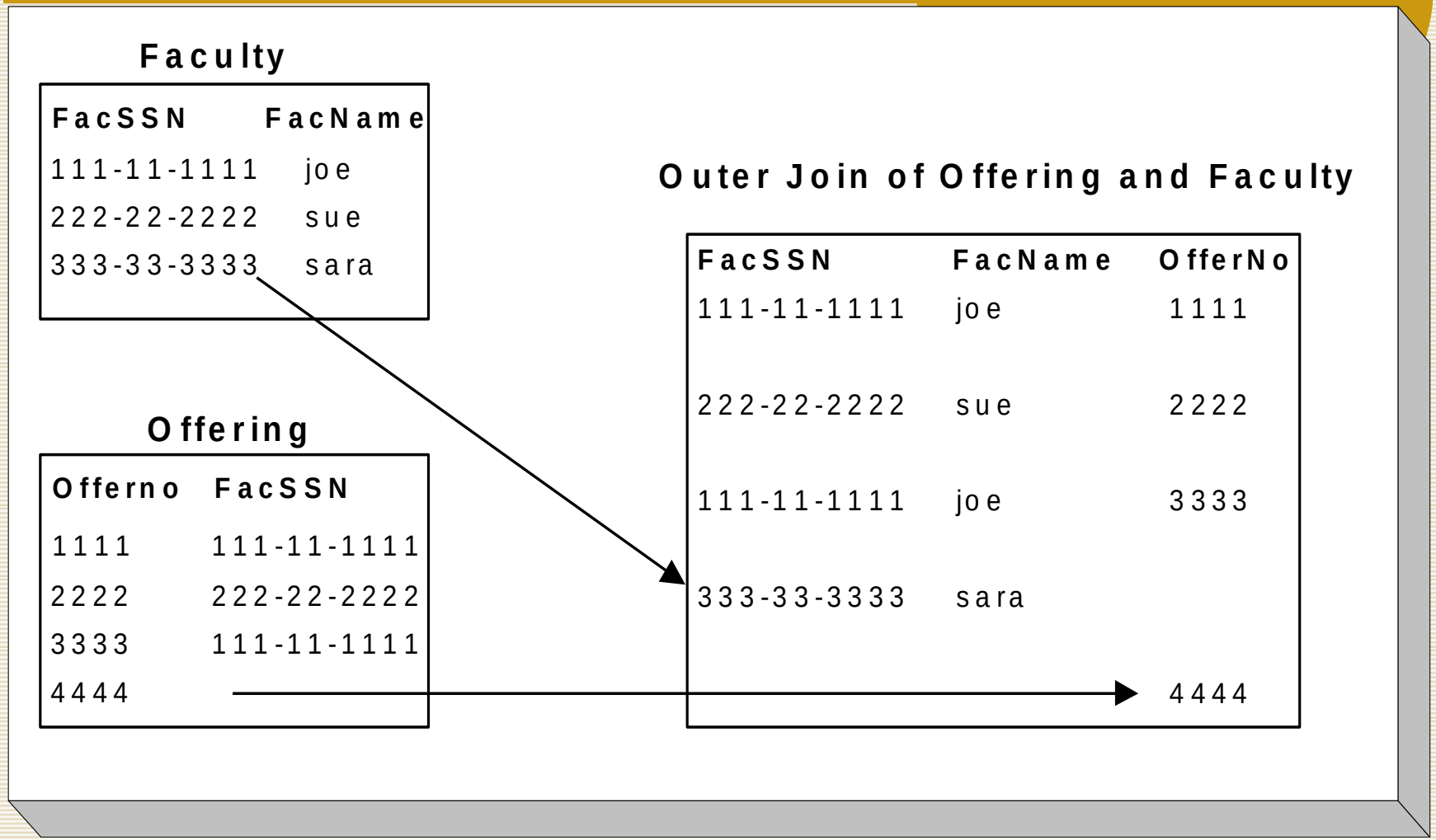
Outer Join Overview

- Join excludes non matching rows
- Preserving non matching rows is important in some business situations
- Outer join variations
  - Full outer join
  - One-sided outer join

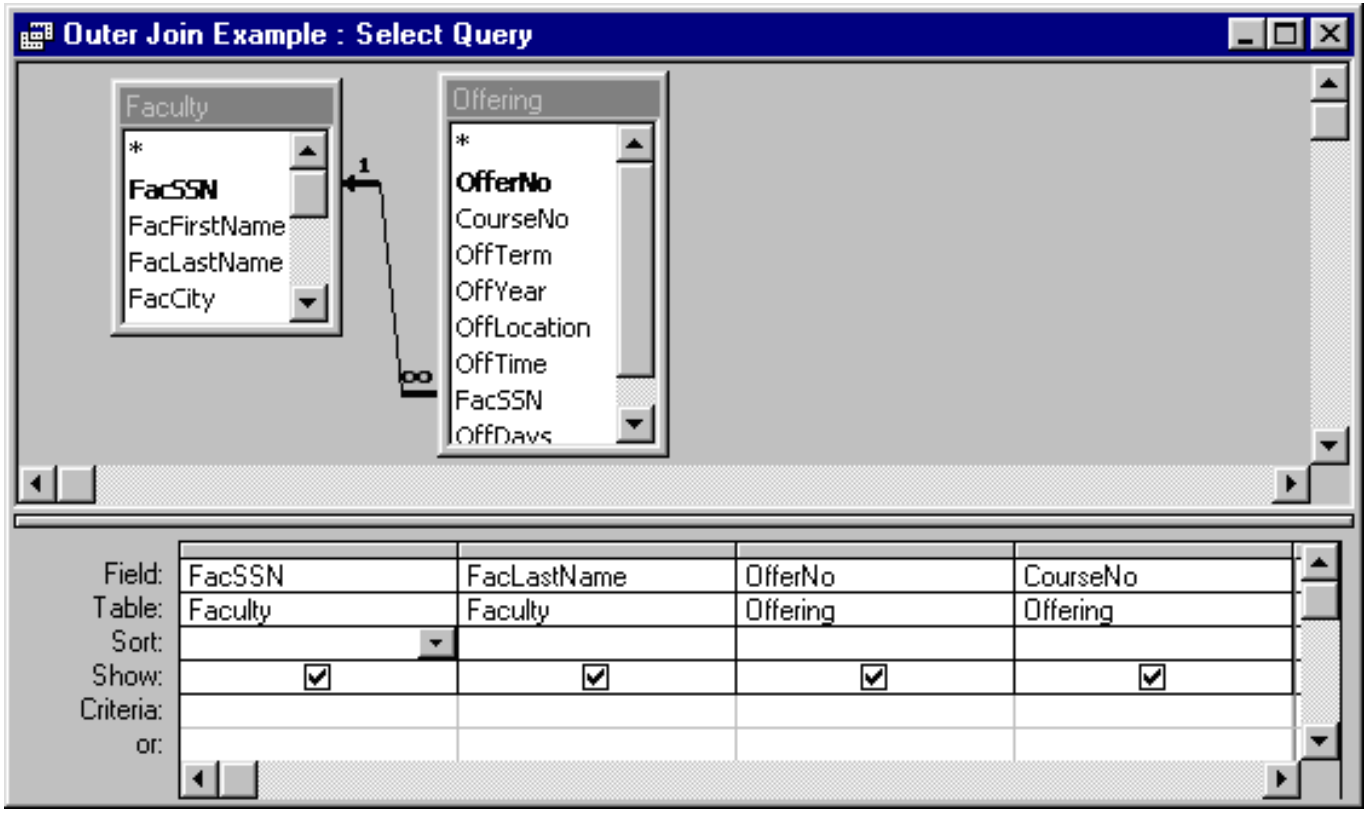
Outer Join Operators



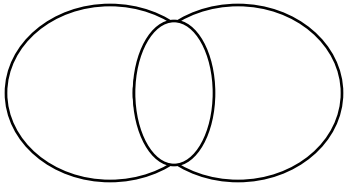
Full Outer Join Example



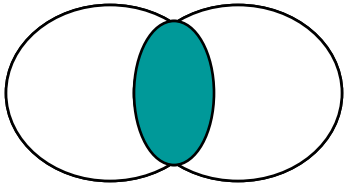
Visual Formulation of Outer Join



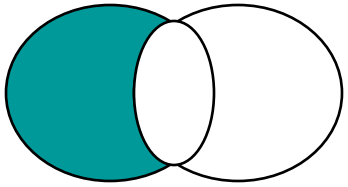
Traditional Set Operators



A UNION B



A INTERSECT B



A MINUS B

Union 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

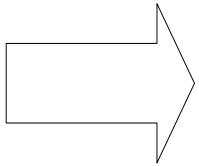
Summarize Operator

- Decision-making operator
- Compresses groups of rows into calculated values
- Simple statistical (aggregate) functions
- Not part of original relational algebra

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



SUMMARIZE Enrollment
ADD AVG(EnrGrade)
GROUP BY StdSSN

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

Divide Operator

- Match on a subset of values
  - Suppliers who supply all parts
  - Faculty who teach every IS course
- Specialized operator
- Typically applied to associative tables representing M-N relationships

Division Example

SuppPart

SuppNo	PartNo
s3	p1
s3	p2
s3	p3
s0	p1
s1	p2

Part

PartNo
p1
p2

SuppPart DIVIDEBY Part

SuppNo
s3

s3 {p1, p2, p3}

contains {p1, p2}

Relational Algebra Summary

Operator	Meaning
Restrict (Select)	Extracts rows that satisfy a specified condition
Project	Extracts specified columns.
Product	Builds a table from two tables consisting of all possible combinations of rows, one from each of the two tables.
Union	Builds a table consisting of all rows appearing in either of two tables
Intersect	Builds a table consisting of all rows appearing in both of two specified tables
Difference	Builds a table consisting of all rows appearing in the first table but not in the second table
Join	Extracts rows from a product of two tables such that two input rows contributing to any output row satisfy some specified condition.
Outer Join	Extracts the matching rows (the join part) of two tables and the "unmatched" rows from both tables.
Divide	Builds a table consisting of all values of one column of a binary (2 column) table that match (in the other column) all values in a unary (1 column) table.
Summarize	Organizes a table on specified grouping columns. Specified aggregate computations are made on each value of the grouping columns.

Summary

- Relational model is commercially dominant
- Learn primary keys, data types, and foreign keys
- Visualize relationships
- Understanding existing databases is crucial to query formulation