Chapter 4

Query Formulation with SQL

Outline

- Background
- Getting started
- Joining tables
- Summarizing tables
- Problem solving guidelines
- Advanced problems
- Data manipulation statements

What is SQL?

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

SQL Standardization History

Year	Features
1986	Core SQL
1989	Integrity constraints
1999	Procedural features
2003	XML features
…	…
2019	Current standard

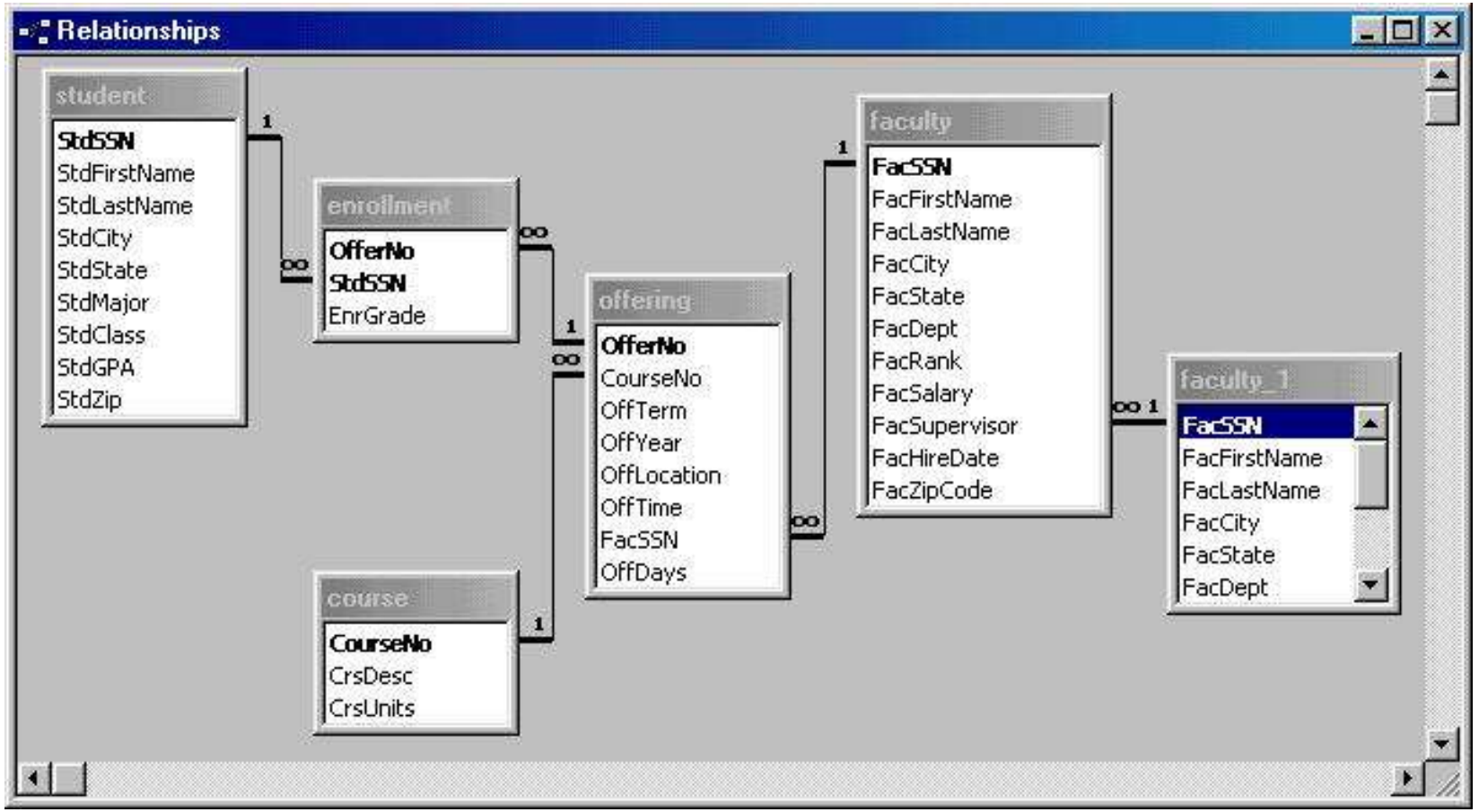
SQL Conformance

- No official conformance testing
- Reasonable conformance on Core SQL
- Large variance on conformance outside of Core SQL
- Difficult to write portable SQL code outside of Core SQL

SELECT Statement Overview

```
SELECT <columns>
  FROM <tables>
  WHERE <conditions>
  GROUP BY <columns>
  HAVING <conditions>
  ORDER BY <sorting specifications>
```

University Database



First SELECT Examples

Example 1

```
SELECT * FROM Faculty
```

Example 2

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

Example 3

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

Example 4

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

Case Sensitivity

- Databases vary on case sensitivity with respect to
  - Table and column names
  - String comparisons

Inexact Matching

- Match against a pattern: LIKE / NOT LIKE operators
- Use meta characters to specify patterns
  - Wildcard (%)
  - Any single character (? or _)

Example

```
SELECT *
 FROM Offering
 WHERE FacFullName LIKE '%Higg%'
```

Using Dates

- Date constants and functions are not standard
- Many databases use 'YYYY-MM-DD' format

Example

```
SELECT FacFirstName, FacLastName, FacHireDate
 FROM Faculty
 WHERE FacHireDate >= '1999-01-01'
   AND FacHireDate <= '2000-12-31'
```

Example

```
SELECT FacFirstName, FacLastName, FacHireDate
 FROM Faculty
 WHERE FacHireDate BETWEEN '1999-01-01'
   AND '2000-12-31'
```

Other Single Table Examples

Example: Testing for null values

```
SELECT OfferNo, CourseNo
 FROM Offering
 WHERE FacSSN IS NULL AND OffTerm = 'SUMMER'
   AND OffYear = 2006
```

Example: Mixing AND and OR

```
SELECT OfferNo, CourseNo, FacSSN
 FROM Offering
 WHERE (OffTerm = 'FALL' AND OffYear = 2005)
    OR (OffTerm = 'WINTER' AND OffYear = 2006)
```

Column Expressions

Example: Math calculations

```
SELECT FacFirstName, FacLastName, FacCity,
       FacSalary*1.1 AS IncreasedSalary,
       FacHireDate
 FROM Faculty
```

Example: String concatenation

```
SELECT FacFirstName || ' ' || FacLastName
          AS FacName
 FROM Faculty
```

Summarizing Tables

- Row summaries important for decision-making tasks
- Summaries occur when SELECT uses aggregate functions:
  - COUNT(), SUM(), MAX(), MIN(), AVG()
- Several rows in the table yield a single row of output

Summary Examples

Example: Summarizing data in entire table

```
SELECT COUNT(FacSSN) AS NumFaculty,
    AVG(FacSalary) AS AvgSalary
 FROM Faculty
```

NumFaculty	AvgSalary
6	67500

GROUP BY Examples

Example: Grouping on a single column

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

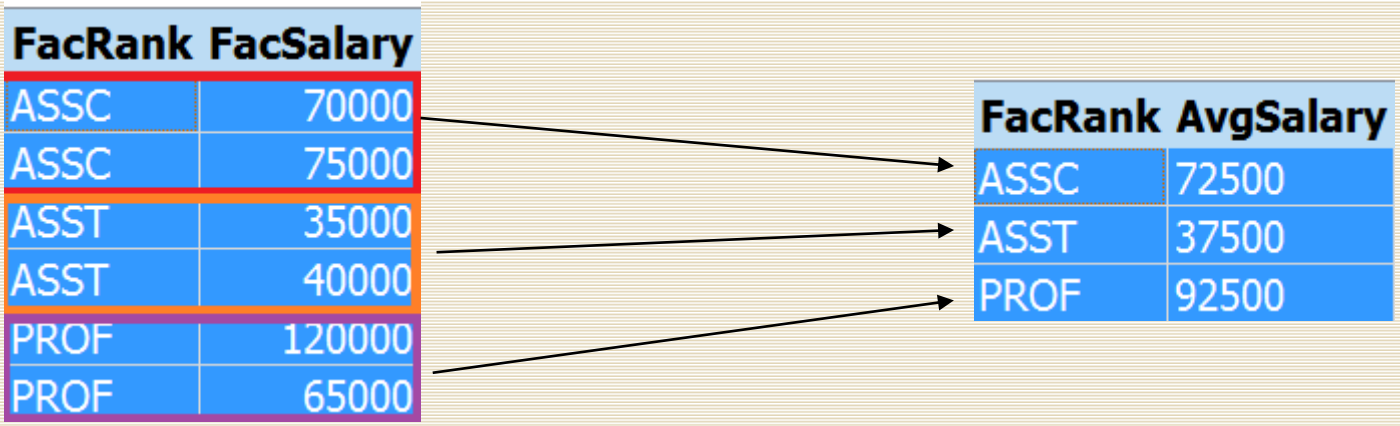
FacRank	AvgSalary
ASSC	72500
ASST	37500
PROF	92500

How GROUP BY Works

```
SELECT FacRank, AVG(FacSalary) AS AvgSalary
   FROM Faculty
   GROUP BY FacRank
   HAVING AVG(FacSalary) < 80000
```

1. Query processor groups rows, ordered by GROUP BY column

2. Query processor collapses each group into one row, computing aggregate function



GROUP BY Examples

Example: Row and group conditions

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

Notes:

- WHERE conditions applied **before** computing aggregate
- HAVING conditions applied **after** computing aggregate

SQL Summarization Rules

- Columns in SELECT and GROUP BY
  - SELECT: scalar and aggregate columns
  - GROUP BY: list all scalar columns
- WHERE versus HAVING
  - Row conditions in WHERE
  - Group conditions in HAVING

Using DISTINCT

- Consider this query:
  - SELECT FacFirstName
    FROM Faculty
- The results contain duplicates, since some faculty have the same first name
- To eliminate duplicate rows from the result, specify the DISTINCT option:
  - SELECT **DISTINCT** FacFirstName
    FROM Faculty

Joins

Join Operator

- Most databases have many tables
- Combine tables using the join operator
- Each record in one table is combined with matching records from the other
- Specify matching condition
  - Can be any comparison but usually =
  - PK = FK most common join condition
  - Relationship diagram useful when combining tables

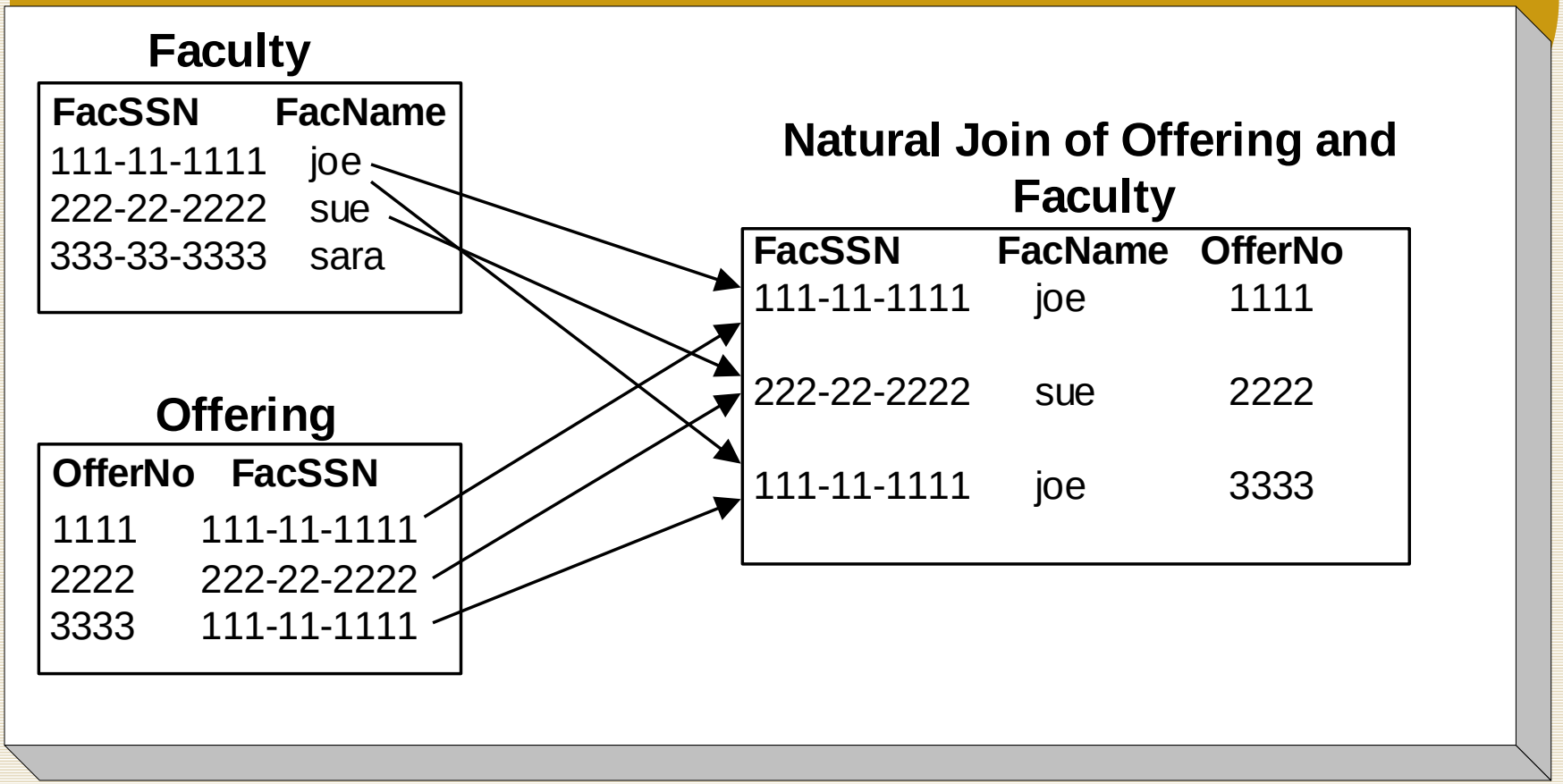
Join Operator Example

- FROM clause contains join operations
- Use INNER JOIN (or just JOIN) and ON keywords

Example

```
SELECT Faculty.FacSSN, FacFirstName,
       OfferNo
 FROM Offering INNER JOIN Faculty ON
        Faculty.FacSSN = Offering.FacSSN
```

Join Example



Alternate Notation

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

Example

```
SELECT Faculty.FacSSN, FacFirstName,
       OfferNo
 FROM Offering, Faculty
 WHERE Faculty.FacSSN = Offering.FacSSN
```

Join Operator Style

Helps distinguish join conditions from non-join conditions

Example 11

```
SELECT OfferNo, CourseNo, FacFirstName,
       FacLastName
 FROM Offering INNER JOIN Faculty
   ON Faculty.FacSSN = Offering.FacSSN
 WHERE OffTerm = 'FALL' AND OffYear = 2007
   AND FacRank = 'ASST' AND CourseNo LIKE 'IS%'
```

Name Qualification

- FacSSN is in both Faculty and Offering
- Must qualify FacSSN with name of table

```
SELECT Faculty.FacSSN,
       FacFirstName, CourseNo
  FROM Faculty JOIN Offering
       ON Faculty.FacSSN = Offering.FacSSN
```

How to Join Two Tables

- The two tables should have a primary key / foreign key relationship
- Put the primary key on one side of the join condition, and the foreign key on the other side

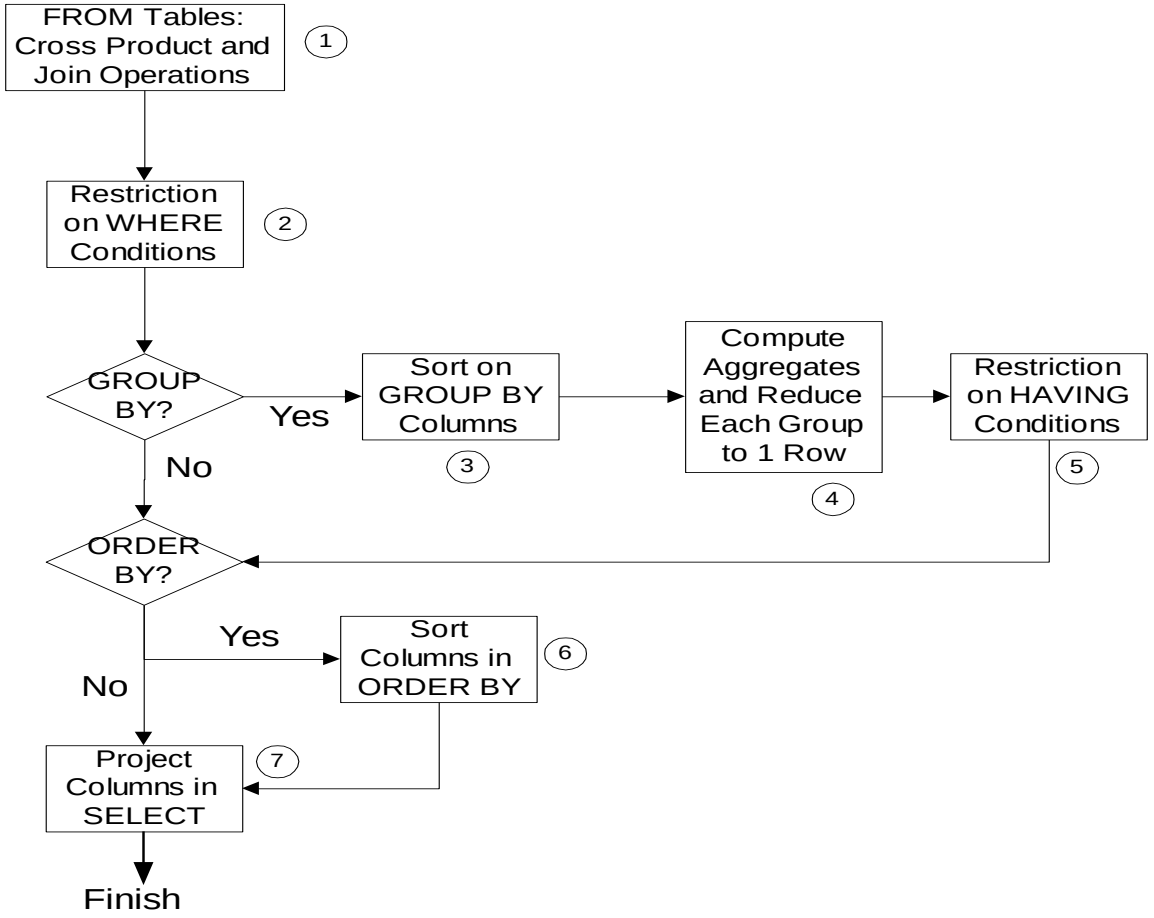
```
SELECT ...
  FROM Faculty JOIN Offering
       ON Faculty.FacSSN = Offering.FacSSN
```

Summarization and Joins

Example 14: List the number of students enrolled in each fall 2007 offering.

```
SELECT Offering.OfferNo,
       COUNT( * ) AS NumStudents
  FROM Enrollment JOIN Offering
       ON Offering.OfferNo = Enrollment.OfferNo
 WHERE OffYear = 2007 AND OffTerm = 'FALL'
 GROUP BY Offering.OfferNo
```

Conceptual Evaluation Process



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 2006 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 = 2007
 GROUP BY FacRank, FacDept
 HAVING COUNT(*) > 1
```

Advanced Problems

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

Joining Three Tables

Example 16: List Leonard Vince's teaching schedule in fall 2005. 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 JOIN Offering
        ON Faculty.FacSSN = Offering.FacSSN
      JOIN Course
        ON Offering.CourseNo = Course.CourseNo
 WHERE
   OffYear = 2007 AND OffTerm = 'FALL'
   AND FacFirstName = 'LEONARD'
   AND FacLastName = 'VINCE'
```

Joining Four Tables

Example 17: List Bob Norbert's course schedule in spring 2006. 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 = 2007 AND OffTerm = 'SPRING'
   AND StdFirstName = 'BOB'
   AND StdLastName = 'NORBERT'
```

Join Two Tables Revisited

- What if the two tables don't have a primary key / foreign key relationship?
- Find a common table that has a primary key / foreign key relationship with both
- Join all three tables

Example: List students and the courses in which they are enrolled

Join Two Tables Revisited

Example: List students and the courses in which they are enrolled

```
SELECT StdFirstName, StdLastName, CourseNo
 FROM Student JOIN Enrollment ON
        Student.StdSSN = Enrollment.StdSSN
        JOIN Offering ON
        Enrollment.OfferNo = Offering.OfferNo
```

Self-Join

- Join a table to itself
- Usually involves a self-referencing relationship
- Requires using a table alias

Self-Join Example

Example 18: List faculty members who have a higher salary than their supervisor. Show the social security number, name, and salary of the faculty and supervisor.

```
SELECT Subr.FacSSN, Subr.FacLastName,
       Subr.FacSalary, Supr.FacSSN,
       Supr.FacLastName, Supr.FacSalary
 FROM Faculty AS Subr INNER JOIN
        Faculty AS Supr
        ON Subr.FacSupervisor = Supr.FacSSN
 WHERE Subr.FacSalary > Supr.FacSalary
```

Summary

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

Further Reading

- SQL by Example, by John Russo https://ebookcentral.proquest.com/lib/bju/detail.action?docID=5602384