

Chapter 4

Query Formulation with SQL

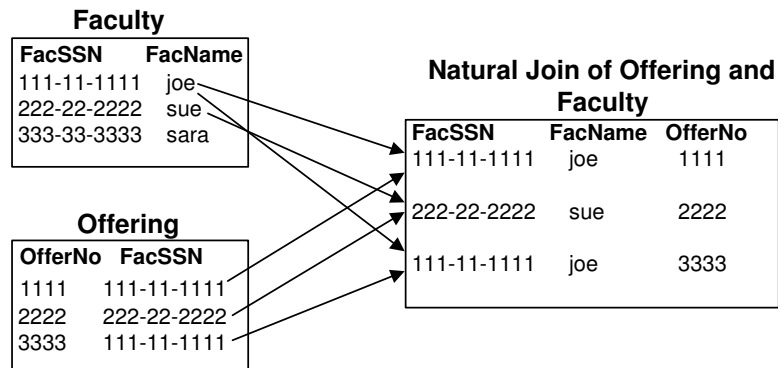
McGraw-Hill/Irwin

Copyright © 2007 by The McGraw-Hill Companies, Inc. All rights reserved.

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



4-3

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 = 2005  
      AND FacRank = 'ASST' AND CourseNo LIKE 'IS*'
```

4-4

Cross Product Style

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

Example 10 (Access)

```
SELECT OfferNo, CourseNo, FacFirstName,  
       FacLastName  
FROM Offering, Faculty  
WHERE OffTerm = 'FALL' AND OffYear = 2005  
      AND FacRank = 'ASST' AND CourseNo LIKE 'IS*'  
      AND Faculty.FacSSN = Offering.FacSSN
```

4-5

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
- Readability versus writability

4-6

Summarizing Tables

- Row summaries important for decision-making tasks
- Row summary
 - Result contains statistical (aggregate) functions
 - Conditions involve statistical functions
- SQL keywords
 - Aggregate functions in the output list
 - GROUP BY: summary columns
 - HAVING: summary conditions

4-7

GROUP BY Examples

Example 12: Grouping on a single column

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

Example 13: 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
```

4-8

SQL Summarization Rules

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

4-9

Summarization and Joins

- Powerful combination
- List join conditions in the WHERE clause

Example 14: List the number of students enrolled in each 2006 offering.

```
SELECT Offering.OfferNo,  
       COUNT(*) AS NumStudents  
FROM Enrollment, Offering  
WHERE Offering.OfferNo = Enrollment.OfferNo  
      AND OffYear = 2006  
GROUP BY Offering.OfferNo
```

4-10

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

4-11

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, Course, Offering
WHERE Faculty.FacSSN = Offering.FacSSN
      AND Offering.CourseNo = Course.CourseNo
      AND OffYear = 2005 AND OffTerm = 'FALL'
      AND FacFirstName = 'Leonard'
      AND FacLastName = 'Vince'
```

4-12

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 = 2006 AND OffTerm = 'SPRING'  
      AND StdFirstName = 'BOB'  
      AND StdLastName = 'NORBERT'
```

4-13

Self-Join

- Join a table to itself
- Usually involve a self-referencing relationship
- 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

4-14

Self-Join Example

Example 18: List faculty members who have a higher salary than their supervisor. List 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 Subr, Faculty Supr  
WHERE Subr.FacSupervisor = Supr.FacSSN  
      AND Subr.FacSalary > Supr.FacSalary
```

4-15

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 in 2006.

```
SELECT FacFirstName, FacLastName, O1.CourseNo  
FROM Faculty, Offering O1, Offering O2  
WHERE Faculty.FacSSN = O1.FacSSN  
      AND Faculty.FacSupervisor = O2.FacSSN  
      AND O1.OffYear = 2006 AND O2.OffYear = 2006  
      AND O1.CourseNo = O2.CourseNo
```

4-16

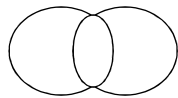
Multiple Column Grouping

Example 20: List the course number, the offering number, and the number of students enrolled. Only include courses offered in spring 2006.

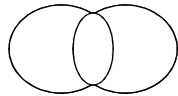
```
SELECT CourseNo, Enrollment.OfferNo,  
       Count(*) AS NumStudents  
FROM Offering, Enrollment  
WHERE Offering.OfferNo = Enrollment.OfferNo  
      AND OffYear = 2006 AND OffTerm = 'SPRING'  
GROUP BY Enrollment.OfferNo, CourseNo
```

4-17

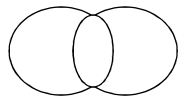
Traditional Set Operators



A UNION B



A INTERSECT B



A MINUS B

4-18

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

4-19

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
```

4-20

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
```

4-21

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
```

4-22

Data Manipulation Statements

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

4-23

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)
```

4-24

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'
```

4-25

DELETE Example

Example 26: Delete all IS majors who are seniors.

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

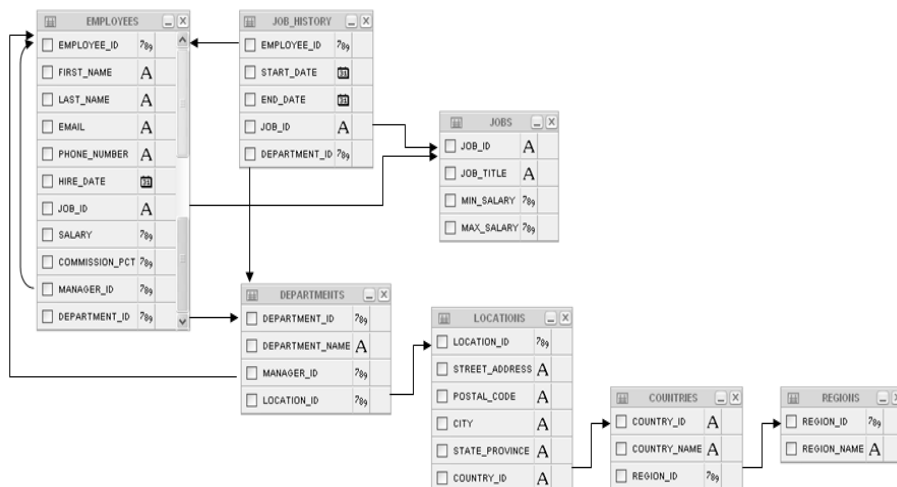
4-26

Summary

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

4-27

HR Database



4-28

Latihan 1

Gunakan database HR di Oracle 10g XE.

Tuliskan SQL statement untuk menampilkan output query berikut ini :

1. Nama lengkap pegawai yang tidak memiliki komisi
2. Jenis pekerjaan yang selisih gaji terbesar & gaji terkecilnya di bawah \$5000
3. Nama lengkap pegawai yang diangkat pada bulan Agustus 1994
4. Nama kota di Inggris yang menjadi lokasi kantor departemen
5. Daftar nama dan alamat seluruh departemen beserta nama lengkap manajernya (nama + alamat departemen, nama manajer)
6. Daftar pegawai (nama, gaji) beserta manajernya (nama, gaji)
7. Daftar riwayat pekerjaan (nama pegawai, nama pekerjaan, periode kerja) khusus untuk pegawai yang gajinya di atas \$10000
8. Daftar rata-rata gaji pegawai tiap departemen (nama departemen, rata-rata gaji)

4-29

Latihan 1

9. Daftar negara beserta jumlah departemen yang berkantor di setiap negara tsb (nama negara, jumlah departemen)
10. Daftar jumlah pegawai untuk setiap jenis pekerjaan (jenis pekerjaan, jumlah pegawai)
11. Daftar persentase komisi tertinggi untuk setiap departemen (nama departemen, komisi tertinggi)
12. Daftar persentase komisi terendah untuk setiap jenis pekerjaan (jenis pekerjaan, komisi terendah)

4-30