Chapter 2

Query Formulation with SQL

What is SQL?

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

DML * DDL * DCL

- Data Manipulation Language (DML) :
  INSERT, UPDATE, DELETE, SELECT
- Data Definition Language (DDL) :
  CREATE DATABASE, DROP DATABASE,
  CREATE TABLE, DROP TABLE, ALTER TABLE
- Data Control Language (DCL) :
  GRANT, REVOKE, COMMIT, ROLLBACK, ETC

4-3

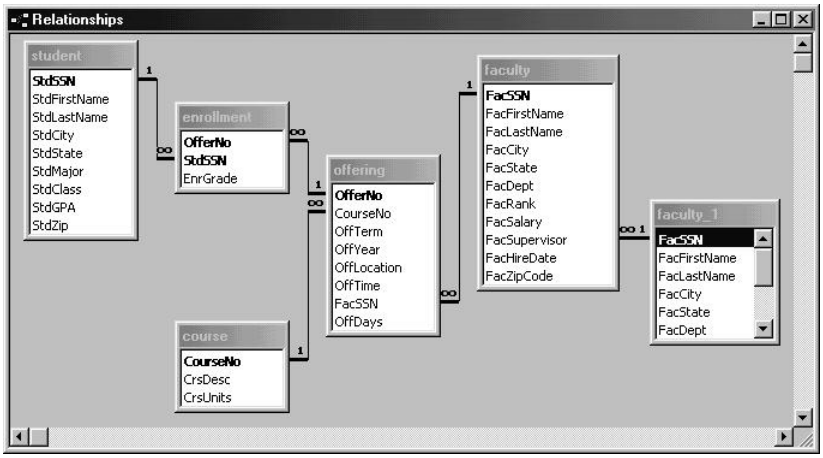
SELECT Statement Overview

`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

4-4

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'
```

Using Expressions

Example 5 (Access)

```
SELECT FacFirstName, FacLastName, FacCity,
       FacSalary*1.1 AS IncreasedSalary,
       FacHireDate
FROM Faculty
WHERE year(FacHireDate) > 1996
```

Example 5 (Oracle)

```
SELECT FacFirstName, FacLastName, FacCity,
      FacSalary*1.1 AS IncreasedSalary,
      FacHireDate
FROM Faculty
WHERE to_number(to_char(FacHireDate, 'YYYY'))
        > 1996
```

4-7

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 'IS*'
```

Example 6 (Oracle)

```
SELECT *
 FROM Offering
 WHERE CourseNo LIKE 'IS%'
```

4-8

Using Dates

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

Example 7 (Access)

```
SELECT FacFirstName, FacLastName, FacHireDate
 FROM Faculty
 WHERE FacHireDate BETWEEN #1/1/1999#
   AND #12/31/2000#
```

Example 7 (Oracle)

```
SELECT FacFirstName, FacLastName, FacHireDate
 FROM Faculty
 WHERE FacHireDate BETWEEN '1-Jan-1999'
   AND '31-Dec-2000'
```

4-9

Other Single Table Examples

Example 8: Testing for null values

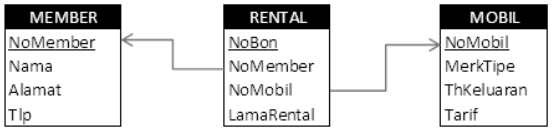
```
 SELECT OfferNo, CourseNo
  FROM Offering
  WHERE FacSSN IS NULL AND OffTerm = 'SUMMER'
    AND OffYear = 2006
```

Example 9: Mixing AND and OR

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

4-10

Inner Join
(Cross Product Style)

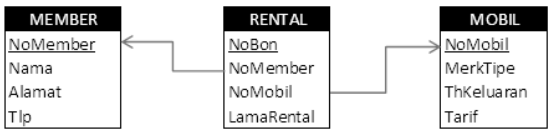



```
SELECT R.NoBon, M.Nama, M.Alamat, R.NoMobil, C.MerkTipe,
R.LamaRental
  FROM Member M, Rental R, Mobil C
  WHERE (M.NoMember = R.NoMember) AND (R.NoMobil = C.NoMobil)

SELECT NoBon, Nama, Alamat, R.NoMobil, MerkTipe, LamaRental
  FROM Member M, Rental R, Mobil C
  WHERE (M.NoMember = R.NoMember) AND (R.NoMobil = C.NoMobil)
```

4-11

Inner Join
(Inner Join Operator Style)




```
SELECT R.NoBon, M.Nama, M.Alamat, R.NoMobil, C.MerkTipe,
R.LamaRental
  FROM Member M INNER JOIN Rental R ON M.NoMember = R.NoMember
               INNER JOIN Mobil C ON R.NoMobil = C.NoMobil

SELECT NoBon, Nama, Alamat, R.NoMobil, MerkTipe, LamaRental
  FROM Member M INNER JOIN Rental R ON M.NoMember = R.NoMember
               INNER JOIN Mobil C ON R.NoMobil = C.NoMobil
```

4-12

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

4-13

Self-Join Example

Example : 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
```

4-14

Fungsi Agregat

```
SELECT SUM(UnitTerjual) AS JumlahTerjual
   FROM DETILJUAL

SELECT AVG(NilaiUas) AS RataRataNilaiUas
   FROM NILAI

SELECT MAX(Harga) AS HargaProdukTermahal
   FROM PRODUK

SELECT MIN(Harga) AS HargaProdukTermurah
   FROM PRODUK

SELECT COUNT(*) AS JumlahPelanggan
   FROM PELANGGAN

SELECT STD(NilaiTugas) AS StandarDeviasiNilaiTugas
   FROM NILAI
```


Group By (1)

MAHASISWA

NPM	NamaMhs	Fakultas	JenisKelamin
2009101	Yunika	MIPA	P
2009102	Yovita	Sastra	P
2009103	Yohanes	Sastra	L
2009104	Yohana	MIPA	P
....			
2010503	Yokohama	Ekonomi	L

[a] *Menampilkan jumlah mahasiswa tiap fakultas :*

```
SELECT Fakultas, COUNT (*) AS JumlahMhs
   FROM MAHASISWA
   GROUP BY Fakultas
```

[a]

Fakultas	JumlahMhs
Ekonomi	90
Sastra	65
MIPA	49

[b] *Menampilkan jumlah mahasiswa per fakultas untuk tiap jenis kelamin :*

```
SELECT Fakultas, JenisKelamin, COUNT (*) AS
JumlahMhs
   FROM MAHASISWA
   GROUP BY Fakultas, JenisKelamin
```

[b]

Fakultas	JenisKelamin	JumlahMhs
Ekonomi	L	35
Ekonomi	P	55
Sastra	L	32
Sastra	P	33
MIPA	L	30
MIPA	P	19

Group By (2)

PRODUK

KdProduk	Merk	Kategori	Harga	Stok
M101	Logitech	Mouse	90000	20
M102	Genius	Mouse	35000	15
F201	EasyDisk	FlashDisk	82500	12
K104	Diamond	Keyboard	65000	23
....				
F222	Apacer	FlashDisk	120000	10

[c] *Menampilkan total stok masing-masing kategori produk :*

```
SELECT Kategori, SUM (Stok) AS TotalStok
   FROM PRODUK
   GROUP BY Kategori
```

[c]

Kategori	TotalStok
Mouse	124
Keyboard	75
Flash Disk	96

[d] *Menampilkan harga tertinggi tiap merk produk :*

```
SELECT Merk, MAX (Harga) AS HargaTertinggi
   FROM PRODUK
   GROUP BY Merk
```

[d]

Merk	HargaTertinggi
Apacer	245000
Diamond	75000
Easy Disk	82500
Genius	115000
Logitech	190000

4-17

Group By & Having

PRODUK

KdProduk	Merk	Kategori	Harga	Stok
M101	Logitech	Mouse	90000	20
M102	Genius	Mouse	35000	15
F201	EasyDisk	FlashDisk	82500	12
K104	Diamond	Keyboard	65000	23
....				
F222	Apacer	FlashDisk	120000	10

[e] *Menampilkan total stok masing-masing kategori selain produk Flash Disk :*

```
SELECT Kategori, SUM (Stok) AS TotalStok
   FROM Produk GROUP BY Kategori
   HAVING Kategori <> 'Flash Disk'
```

[e]

Kategori	TotalStok
Mouse	124
Keyboard	75

[f] *Menampilkan harga tertinggi untuk merk Apacer, Diamond & Genius :*

```
SELECT Merk, MAX (Harga) AS HargaTertinggi
   FROM Produk GROUP BY Merk
   HAVING Merk IN ('Apacer', 'Diamond', 'Genius')
```

[f]

Merk	HargaTertinggi
Apacer	245000
Diamond	75000
Genius	115000

4-18

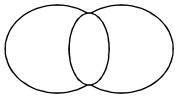
Nested Query

Example : List finance faculty who teach IS courses.

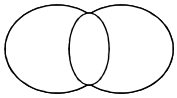
```
SELECT FacSSN, FacLastName, FacDept
 FROM Faculty
 WHERE FacDept = 'FIN' AND FacSSN IN
  (SELECT FacSSN FROM Offering
     WHERE CourseNo LIKE 'IS*')
```

4-19

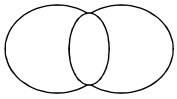
Traditional Set Operators



A UNION B



A INTERSECT B



A MINUS B

4-20

Union Compatibility

- Strong requirement
  - Same number of columns
  - Each corresponding column is compatible
  - Positional correspondence

4-21

SQL UNION Example

Example : 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
```

4-22

Oracle INTERSECT Example

Example : 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
```

4-23

Oracle MINUS Example

Example : 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
```

4-24

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

4-25

INSERT Example

Example : 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)
```

4-26

UPDATE Example

Example : Change the major and class of Homer Wells.

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

4-27

DELETE Example

Example : Delete all IS majors who are seniors.

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

4-28