David M. Kroenke and David J. Auer

Database Processing:

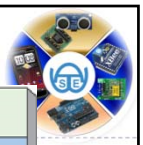
Fundamentals, Design, and Implementation

Chapter Two:

Introduction to Structured Query Language

Chapter Objectives II

- Built-in functions of SUM, COUNT, MIN, MAX, and AVG
- GROUP BY clause
- HAVING clause
- retrieve data from multiple tables
  - Using SUBQUERY
  - Using JOIN
  - Set operations AND、OR、NOT



Extracted Retail Sales Data Format

Table	Column	Date Type
RETAIL_ORDER	OrderNumber	Integer
	StoreNumber	Integer
	StoreZip	Character (9)
	OrderMonth	Character (12)
	OrderYear	Integer
	OrderTotal	Currency
ORDER_ITEM	OrderNumber	Integer
	SKU	Integer
	Quantity	Integer
	Price	Currency
	ExtendedPrice	Currency
SKU_DATA	SKU	Integer
	SKU_Description	Character (35)
	Department	Character (30)
	Buyer	Character (30)

The SQL SELECT Statement

- **The fundamental framework for SQL query states is the SQL SELECT statement.**
  - **SELECT {ColumnName(s)}**
  - **FROM {TableName(s)}**
  - **WHERE {Conditions} ;**

WHERE Clause Options— LIKE and Wildcards

- The SQL keyword LIKE can be combined with wildcard symbols:

  _ = exactly one character

  % = any set of one or more characters

WHERE Clause Options— LIKE and Wildcards

```
SELECT    *
FROM SKU_DATA
WHERE     Buyer LIKE 'Pete%';
```

SKU	SKU_Description	Department	Buyer
100100	Std. Scuba Tank, Yellow	Water Sports	Pete Hansen
100200	Std. Scuba Tank, Magenta	Water Sports	Pete Hansen
101100	Dive Mask, Small Clear	Water Sports	Nancy Meyers
101200	Dive Mask, Med Clear	Water Sports	Nancy Meyers
201000	Half-dome Tent	Camping	Cindy Lo
202000	Half-dome Tent Footprint	Camping	Cindy Lo
301000	Light Fly Climbing Harness	Climbing	Jerry Martin
302000	Locking carabiner, Oval	Climbing	Jerry Martin

WHERE Clause Options— LIKE and Wildcards

```
SELECT    *
FROM SKU_DATA
WHERE     SKU_Description LIKE '%Tent%';
```

SKU	SKU_Description	Department	Buyer
100100	Std. Scuba Tank, Yellow	Water Sports	Pete Hansen
100200	Std. Scuba Tank, Magenta	Water Sports	Pete Hansen
101100	Dive Mask, Small Clear	Water Sports	Nancy Meyers
101200	Dive Mask, Med Clear	Water Sports	Nancy Meyers
201000	Half-dome Tent	Camping	Cindy Lo
202000	Half-dome Tent Footprint	Camping	Cindy Lo
301000	Light Fly Climbing Harness	Climbing	Jerry Martin
302000	Locking carabiner, Oval	Climbing	Jerry Martin

WHERE Clause Options— LIKE and Wildcards

```
SELECT    *
FROM SKU_DATA
WHERE     SKU LIKE '%2__';
```

SKU	SKU_Description	Department	Buyer
100100	Std. Scuba Tank, Yellow	Water Sports	Pete Hansen
100200	Std. Scuba Tank, Magenta	Water Sports	Pete Hansen
101100	Dive Mask, Small Clear	Water Sports	Nancy Meyers
101200	Dive Mask, Med Clear	Water Sports	Nancy Meyers
201000	Half-dome Tent	Camping	Cindy Lo
202000	Half-dome Tent Footprint	Camping	Cindy Lo
301000	Light Fly Climbing Harness	Climbing	Jerry Martin
302000	Locking carabiner, Oval	Climbing	Jerry Martin

驗收時間

- 作業檢討
- 開考~~~

SQL Built-In Functions

- There are five SQL built-in functions:
  - COUNT
  - SUM
  - AVG
  - MIN
  - MAX

SQL Built-In Functions

```
SELECT   SUM (OrderTotal)
FROM     RETAIL_ORDER;
```

OrderNumber	StoreNumber	StoreZip	OrderMonth	OrderYear	OrderTota
1000	10	98110	December	2008	445.0
2000	20	02335	December	2008	310.0
3000	10	98110	January	2009	480.0

	(No colimn name)
1	1235.00

SQL Built-In Functions

```
SELECT   SUM (OrderTotal)
         AS   OrderSum
FROM     RETAIL_ORDER;
```

OrderNumber	StoreNumber	StoreZip	OrderMonth	OrderYear	OrderTotal
1000	10	98110	December	2008	445.00
2000	20	02335	December	2008	310.00
3000	10	98110	January	2009	480.00

	OrderSum
1	1235.00

SQL Built-In Functions

```
SELECT   SUM (ExtendedPrice)
         AS   Order3000Sum
FROM     ORDER_ITEM
WHERE    OrderNumber = 3000;
```

OrderNumber	SKU	Quantity	Price	ExtendedPrice
3000	100200	1	300.00	300.00
2000	101100	4	50.00	200.00
3000	101100	2	50.00	100.00
2000	101200	2	50.00	100.00
3000	101200	1	50.00	50.00
1000	201000	1	300.00	300.00
1000	202000	1	130.00	130.00

SQL Built-In Functions

```
SELECT SUM (ExtendedPrice) AS OrderItemSum,
       AVG (ExtendedPrice) AS OrderItemAvg,
       MIN (ExtendedPrice) AS OrderItemMin,
        MAX (ExtendedPrice) AS OrderItemMax
FROM    ORDER_ITEM;
```

OrderNumber	SKU	Quantity	Price	ExtendedPrice
3000	100200	1	300.00	300.00
2000	101100	4	50.00	200.00
3000	101100	2	50.00	100.00
2000	101200	2	50.00	100.00
3000	101200	1	50.00	50.00
1000	201000	1	300.00	300.00
1000	202000	1	130.00	130.00

SQL Built-In Functions

```
SELECT   COUNT(*) AS NumberOfRows
FROM     ORDER_ITEM;
```

OrderNumber	SKU	Quantity	Price	ExtendedPrice
3000	100200	1	300.00	300.00
2000	101100	4	50.00	200.00
3000	101100	2	50.00	100.00
2000	101200	2	50.00	100.00
3000	101200	1	50.00	50.00
1000	201000	1	300.00	300.00
1000	202000	1	130.00	130.00

SQL Built-In Functions

```
SELECT   COUNT(Department)AS DeptCount
FROM     SKU_DATA;
```

?與前面的 count(*)有何不同?

?我會想要個數?還是種類?

SKU	SKU_Description	Department	Buyer
100100	Std. Scuba Tank, Yellow	Water Sports	Pete Hansen
100200	Std. Scuba Tank, Magenta	Water Sports	Pete Hansen
101100	Dive Mask, Small Clear	Water Sports	Nancy Meyers
101200	Dive Mask, Med Clear	Water Sports	Nancy Meyers
201000	Half-dome Tent	Camping	Cindy Lo
202000	Half-dome Tent Footprint	Camping	Cindy Lo
301000	Light Fly Climbing Harness	Climbing	Jerry Martin
302000	Locking carabiner, Oval	Climbing	Jerry Martin

SQL Built-In Functions

```
SELECT   COUNT(DISTINCT Department)
         AS DeptCount
FROM     SKU_DATA;
```

SKU	SKU_Description	Department	Buyer
100100	Std. Scuba Tank, Yellow	Water Sports	Pete Hansen
100200	Std. Scuba Tank, Magenta	Water Sports	Pete Hansen
101100	Dive Mask, Small Clear	Water Sports	Nancy Meyers
101200	Dive Mask, Med Clear	Water Sports	Nancy Meyers
201000	Half-dome Tent	Camping	Cindy Lo
202000	Half-dome Tent Footprint	Camping	Cindy Lo
301000	Light Fly Climbing Harness	Climbing	Jerry Martin
302000	Locking carabiner, Oval	Climbing	Jerry Martin

Limitations to built-in function

- Combine with a table column name

```
SELECT   Department, COUNT(*)
FROM     SKU_DATA;
```

SKU	SKU_Description	Department	Buyer
100100	Std. Scuba Tank, Yellow	Water Sports	Pete Hansen
100200	Std. Scuba Tank, Magenta	Water Sports	Pete Hansen
101100	Dive Mask, Small Clear	Water Sports	Nancy Meyers
101200	Dive Mask, Med Clear	Water Sports	Nancy Meyers
201000	Half-dome Tent	Camping	Cindy Lo
202000	Half-dome Tent Footprint	Camping	Cindy Lo
301000	Light Fly Climbing Harness	Climbing	Jerry Martin
302000	Locking carabiner, Oval	Climbing	Jerry Martin

The SQL Keyword GROUP BY

```
SELECT   Department, COUNT(*) AS
         Dept_SKU_Count
FROM     SKU_DATA
GROUP BY     Department;
```

SKU	SKU_Description	Department	Buyer
100100	Std. Scuba Tank, Yellow	Water Sports	Pete Hansen
100200	Std. Scuba Tank, Magenta	Water Sports	Pete Hansen
101100	Dive Mask, Small Clear	Water Sports	Nancy Meyers
101200	Dive Mask, Med Clear	Water Sports	Nancy Meyers
201000	Half-dome Tent	Camping	Cindy Lo
202000	Half-dome Tent Footprint	Camping	Cindy Lo
301000	Light Fly Climbing Harness	Climbing	Jerry Martin
302000	Locking carabiner, Oval	Climbing	Jerry Martin

The SQL Keyword GROUP BY

```
SELECT   Department, Buyer,COUNT(*) AS
         Dept_Buyer_SKU_Count
FROM     SKU_DATA
GROUP BY     Department, Buyer;
```

SKU	SKU_Description	Department	Buyer
100100	Std. Scuba Tank, Yellow	Water Sports	Pete Hansen
100200	Std. Scuba Tank, Magenta	Water Sports	Pete Hansen
101100	Dive Mask, Small Clear	Water Sports	Nancy Meyers
101200	Dive Mask, Med Clear	Water Sports	Nancy Meyers
201000	Half-dome Tent	Camping	Cindy Lo
202000	Half-dome Tent Footprint	Camping	Cindy Lo
301000	Light Fly Climbing Harness	Climbing	Jerry Martin
302000	Locking carabiner, Oval	Climbing	Jerry Martin

The SQL Keyword GROUP BY

```
SELECT   Department, Buyer,
         COUNT(*) AS
         Dept_Buyer_SKU_Count
FROM     SKU_DATA
GROUP BY     Department;
```


The SQL Keyword GROUP BY

```
SELECT   Department, COUNT(*) AS
         Dept_SKU_Count
FROM     SKU_DATA
WHERE    SKU <> 302000
GROUP BY     Department
ORDER BY     Dept_SKU_Count;
```

SKU	SKU_Description	Department	Buyer
100100	Std. Scuba Tank, Yellow	Water Sports	Pete Hansen
100200	Std. Scuba Tank, Magenta	Water Sports	Pete Hansen
101100	Dive Mask, Small Clear	Water Sports	Nancy Meyers
101200	Dive Mask, Med Clear	Water Sports	Nancy Meyers
201000	Half-dome Tent	Camping	Cindy Lo
202000	Half-dome Tent Footprint	Camping	Cindy Lo
301000	Light Fly Climbing Harness	Climbing	Jerry Martin
302000	Locking carabiner, Oval	Climbing	Jerry Martin

Limitations to built-in function

- Using in WHERE clause

~~SELECT *~~
~~FROM RETAIL_ORDER~~
~~WHERE OrderTotal > AVG(OrderTotal);~~

The SQL Keyword GROUP BY

```
SELECT   Department, COUNT(*) AS
         Dept_SKU_Count
FROM     SKU_DATA
WHERE    SKU <> 302000
GROUP BY      Department
HAVING        COUNT (*) > 1
ORDER BY      Dept_SKU_Count;
```

	Department	Dept_SKU_Count
1	Climbing	1
2	Camping	2
3	Water Sports	4

The SQL Keyword ORDER

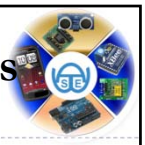
- SELECT
- FROM
- WHERE
- GROUP BY
- HAVING
- ORDER BY

Arithmetic in SELECT Statements

```
SELECT   Quantity * Price AS EP,
         ExtendedPrice
FROM     ORDER_ITEM;
```

OrderNumber	SKU	Quantity	Price	ExtendedPrice
3000	100200	1	300.00	300.00
2000	101100	4	50.00	200.00
3000	101100	2	50.00	100.00
2000	101200	2	50.00	100.00
3000	101200	1	50.00	50.00
1000	201000	1	300.00	300.00
1000	202000	1	130.00	130.00

String Functions in SELECT Statements



```
SELECT   Buyer + ' in ' + Department
         AS Sponsor
FROM     SKU_DATA;
```

SKU	SKU_Description	Department	Buyer
100100	Std. Scuba Tank, Yellow	Water Sports	Pete Hansen
100200	Std. Scuba Tank, Magenta	Water Sports	Pete Hansen
101100	Dive Mask, Small Clear	Water Sports	Nancy Meyers
101200	Dive Mask, Med Clear	Water Sports	Nancy Meyers
201000	Half-dome Tent	Camping	Cindy Lo
202000	Half-dome Tent Footprint	Camping	Cindy Lo
301000	Light Fly Climbing Harness	Climbing	Jerry Martin
302000	Locking carabiner, Oval	Climbing	Jerry Martin

String Functions in SELECT Statements

```
SELECT   DISTINCT RTRIM (Buyer)
         + ' in ' + RTRIM (Department)
         AS Sponsor
FROM     SKU_DATA;
```

SKU	SKU_Description	Department	Buyer
100100	Std. Scuba Tank, Yellow	Water Sports	Pete Hansen
100200	Std. Scuba Tank, Magenta	Water Sports	Pete Hansen
101100	Dive Mask, Small Clear	Water Sports	Nancy Meyers
101200	Dive Mask, Med Clear	Water Sports	Nancy Meyers
201000	Half-dome Tent	Camping	Cindy Lo
202000	Half-dome Tent Footprint	Camping	Cindy Lo
301000	Light Fly Climbing Harness	Climbing	Jerry Martin
302000	Locking carabiner, Oval	Climbing	Jerry Martin

The SQL Keyword ORDER

- SELECT
- FROM
- WHERE
- GROUP BY
- HAVING
- ORDER BY

SELE
Re
FROM
WHER

SKU	SKU_Description	Department	Buyer
100100	Std. Scuba Tank, Yellow	Water Sports	Pete Hansen
100200	Std. Scuba Tank, Magenta	Water Sports	Pete Hansen
101100	Dive Mask, Small Clear	Water Sports	Nancy Meyers
101200	Dive Mask, Med Clear	Water Sports	Nancy Meyers
201000	Half-dome Tent	Camping	Cindy Lo
202000	Half-dome Tent Footprint	Camping	Cindy Lo
301000	Light Fly Climbing Harness	Climbing	Jerry Martin
302000	Locking carabiner, Oval	Climbing	Jerry Martin

OrderNumber	SKU	Quantity	Price	ExtendedPrice
3000	100200	1	300.00	300.00
2000	101100	4	50.00	200.00
3000	101100	2	50.00	100.00
2000	101200	2	50.00	100.00
3000	101200	1	50.00	50.00
1000	201000	1	300.00	300.00
1000	202000	1	130.00	130.00

查詢WaterSport已賣出的總價

試想一

- 如果我要查詢:
某個日期(例如2009年一月)有哪些人買了我們家產品，有沒有辦法查到?

Table	Column	Date Type
RETAIL_ORDER	OrderNumber	Integer
	StoreNumber	Integer
	StoreZip	Character (9)
	OrderMonth	Character (12)
	OrderYear	Integer
	OrderTotal	Currency
ORDER_ITEM	OrderNumber	Integer
	SKU	Integer
	Quantity	Integer
	Price	Currency
	ExtendedPrice	Currency
SKU_DATA	SKU	Integer
	SKU_Description	Character (35)
	Department	Character (30)
	Buyer	Character (30)

2-31

Querying Multiple Tables: Subqueries

```
SELECT Buyer
FROM   SKU_DATA
WHERE  SKU IN
```

	Buyer
1	Pete Hansen
2	Nancy Meyers

SKU	SKU_Description	Department	Buyer
100100	Std. Scuba Tank, Yellow	Water Sports	Pete Hansen
100200	Std. Scuba Tank, Magenta	Water Sports	Pete Hansen
101100	Dive Mask, Small Clear	Water Sports	Nancy Meyers
101200	Dive Mask, Med Clear	Water Sports	Nancy Meyers
201000	Half-dome Tent	Camping	Cindy Lo
202000	Half-dome Tent Footprint	Camping	Cindy Lo
301000	Light Fly Climbing Harness	Climbing	Jerry Martin
302000	Locking carabiner, Oval	Climbing	Jerry Martin

3000	101200	1	30.00	30.00	0.00
1000	201000	1	300.00	300.00	0.00

Querying Multiple Tables with SQL Subqueries

```
/* *** SQL-Query-CH02-57 *** */
SELECT     Buyer, Department, COUNT(SKU) AS Number_Of_SKU_Sold
FROM       SKU_DATA
WHERE      SKU IN
           (SELECT    SKU
            FROM      ORDER_ITEM
            WHERE     OrderNumber  IN
                      (SELECT      OrderNumber
                       FROM        RETAIL_ORDER
                       WHERE       OrderMonth='January'
                         AND       OrderYear=2015))
GROUP BY   Buyer, Department
ORDER BY   Number_Of_SKU_Sold;
```

	Buyer	Department	Number_Of_SKU_Sold
1	Pete Hansen	Water Sports	1
2	Nancy Meyers	Water Sports	2

試想二

- 如果我要查詢:
每位客戶花了多少錢，購買我的產品?

Table	Column	Date Type
RETAIL_ORDER	OrderNumber	Integer
	StoreNumber	Integer
	StoreZip	Character (9)
	OrderMonth	Character (12)
	OrderYear	Integer
	OrderTotal	Currency
ORDER_ITEM	OrderNumber	Integer
	SKU	Integer
	Quantity	Integer
	Price	Currency
	ExtendedPrice	Currency
SKU_DATA	SKU	Integer
	SKU_Description	Character (35)
	Department	Character (30)
	Buyer	Character (30)

Querying Multiple Tables: Join

```
SELECT   Buyer, ExtendedPrice
FROM     SKU_DATA, ORDER_ITEM
WHERE    SKU_DATA.SKU = ORDER_ITEM.SKU;
```

SKU	SKU_Description		Department		Buyer	
100100	Std. Scuba	OrderNumber	SKU	Quantity	Price	ExtendedPrice
100200	Std. Scuba	3000	100200	1	300.00	300.00
101100	Dive Mask	2000	101100	4	50.00	200.00
101200	Dive Mask	3000	101100	2	50.00	100.00
201000	Half-dome	2000	101200	2	50.00	100.00
202000	Half-dome	3000	101200	1	50.00	50.00
301000	Light Fly C	1000	201000	1	300.00	300.00
302000	Locking ca	1000	202000	1	130.00	130.00

同一個名字分很多次出現？給我加好的吧！

Querying Multiple Tables: Joins

```
SELECT    Buyer, SUM(ExtendedPrice)
          AS BuyerRevenue
FROM      SKU_DATA, ORDER_ITEM
WHERE     SKU_DATA.SKU = ORDER_ITEM.SKU
GROUP BY  Buyer
ORDER BY  BuyerRevenue DESC;
```

	Buyer	ExtendedPrice
1	Pete Hansen	300.00
2	Nancy Meyers	200.00
3	Nancy Meyers	100.00
4	Nancy Meyers	100.00
5	Nancy Meyers	50.00
6	Cindy Lo	300.00
7	Cindy Lo	130.00

Querying Multiple Tables: Joins

```
SELECT    Buyer, ExtendedPrice, OrderMonth
FROM      SKU_DATA, ORDER_ITEM, RETAIL_ORDER
WHERE     SKU_DATA.SKU = ORDER_ITEM.SKU
     AND  ORDER_ITEM.OrderNumber =
          RETAIL_ORDER.OrderNumber;
```

OrderNumber	StoreNumber	StoreZip	OrderMonth	OrderYear	OrderTotal

SKU	SKU_Description	Department	Buyer

OrderNumber	SKU	Quantity	Price	ExtendedPrice
3000	100200	1	300.00	300.00
2000	101100	4	50.00	200.00
3000	101100	2	50.00	100.00
2000	101200	2	50.00	100.00

Pete Hansen
Pete Hansen
Nancy Meyers
Nancy Meyers
Cindy Lo
Cindy Lo

Subqueries versus Joins

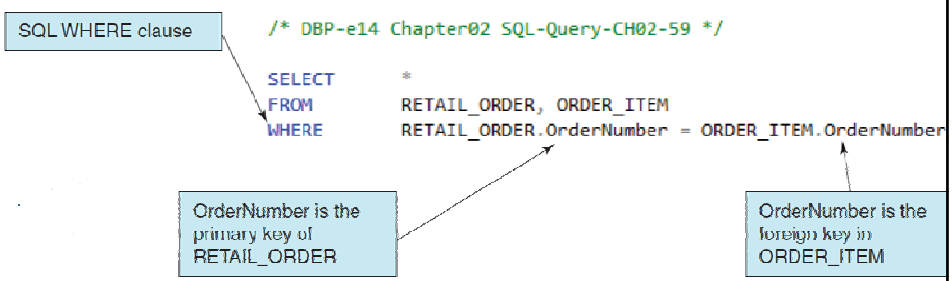
- *Subqueries* and *joins* both process multiple tables.
- A subquery can only be used to retrieve data from the top table.
- A join can be used to obtain data from any number of tables.

Querying Multiple Tables with SQL Joins

- In an SQL join operation, the SQL JOIN operator is used to combine parts or all of two or more tables.
  - Explicit join — the SQL JOIN operator is used as part of the SQL statement.
  - Implicit join — the SQL JOIN operator is *not* used as part of the SQL statement.

Implicit INNER JOIN

- By selecting rows by matching by the primary key values of one table with the foreign key values of a second table, we produce an SQL INNER JOIN.



- Because the SQL JOIN keyword does *not* appear in the SQL statement, this is an implicit join.

Implicit INNER JOIN

```
/* *** SQL-Query-CH02-59 *** */
SELECT      *
FROM        RETAIL_ORDER, ORDER_ITEM
WHERE       RETAIL_ORDER.OrderNumber = ORDER_ITEM.OrderNumber
```

See Page 123

	OrderNumber	StoreNumber	StoreZip	OrderMonth	OrderYear	OrderTotal	OrderNumber	SKU	Quantity	Price	ExtendedPrice
1	3000	10	98110	January	2015	480.00	3000	100200	1	300.00	300.00
2	2000	20	02335	December	2014	310.00	2000	101100	4	50.00	200.00
3	3000	10	98110	January	2015	480.00	3000	101100	2	50.00	100.00
4	2000	20	02335	December	2014	310.00	2000	101200	2	50.00	100.00
5	3000	10	98110	January	2015	480.00	3000	101200	1	50.00	50.00
6	1000	10	98110	December	2014	445.00	1000	201000	1	300.00	300.00
7	1000	10	98110	December	2014	445.00	1000	202000	1	130.00	130.00

Implicit INNER JOIN with ORDER

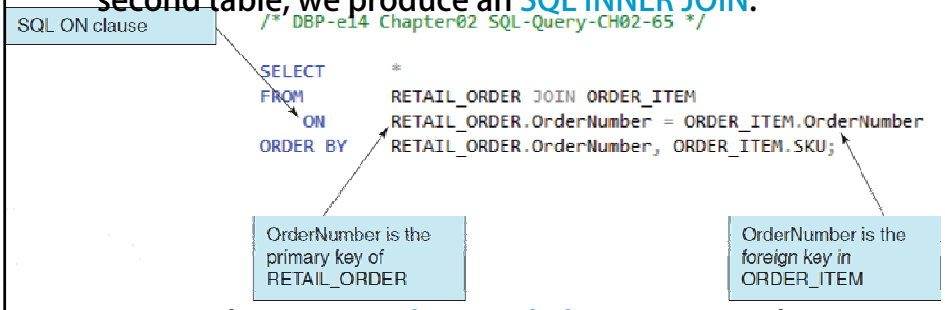
With an SQL ORDER BY clause for easier reading by OrderNumber:

```
/* *** SQL-Query-CH02-60 *** */
SELECT      *
FROM        RETAIL_ORDER, ORDER_ITEM
WHERE       RETAIL_ORDER.OrderNumber=ORDER_ITEM.OrderNumber
ORDER BY    RETAIL_ORDER.OrderNumber, ORDER_ITEM.SKU;
```

	OrderNumber	StoreNumber	StoreZip	OrderMonth	OrderYear	OrderTotal	OrderNumber	SKU	Quantity	Price	ExtendedPrice
1	1000	10	98110	December	2014	445.00	1000	201000	1	300.00	300.00
2	1000	10	98110	December	2014	445.00	1000	202000	1	130.00	130.00
3	2000	20	02335	December	2014	310.00	2000	101100	4	50.00	200.00
4	2000	20	02335	December	2014	310.00	2000	101200	2	50.00	100.00
5	3000	10	98110	January	2015	480.00	3000	100200	1	300.00	300.00
6	3000	10	98110	January	2015	480.00	3000	101100	2	50.00	100.00
7	3000	10	98110	January	2015	480.00	3000	101200	1	50.00	50.00

Explicit SQL INNER JOIN

- By selecting rows by matching by the primary key values of one table with the foreign key values of a second table, we produce an SQL INNER JOIN.



- Because the SQL JOIN keyword *does* appear in the SQL statement, this is an explicit join.

SQL JOIN ON Syntax

- In SQL JOIN ON syntax:
  - The **SQL JOIN keyword** is placed between the table names in the **SQL FROM clause**, where it replaces the comma that previously separated the two table names, and
  - The **SQL ON keyword** now leads into an **SQL ON clause**, which includes the statement of matching key values that was previously in an **SQL WHERE clause**.
  - The **SQL WHERE clause** is *no longer used as part of the join*, which makes it easier to read the actual restrictions on the rows in the query in the **SQL WHERE clause** itself.
- The **explicit SQL JOIN ON syntax** is currently considered as the proper way to write SQL join operations, and the older implicit SQL syntax is considered an archaic, older syntax (but it still works).

SQL JOIN ON Syntax

```
/* *** SQL-Query-CH02-66 *** */
SELECT      *
FROM        RETAIL_ORDER JOIN ORDER_ITEM
      ON    RETAIL_ORDER.OrderNumber = ORDER_ITEM.OrderNumber
WHERE       OrderYear = '2014'
ORDER BY    RETAIL_ORDER.OrderNumber, ORDER_ITEM.SKU;
```

	OrderNumber	StoreNumber	StoreZip	OrderMonth	OrderYear	OrderTotal	OrderNumber	SKU	Quantity	Price	ExtendedPrice
1	1000	10	98110	December	2014	445.00	1000	201000	1	300.00	300.00
2	1000	10	98110	December	2014	445.00	1000	202000	1	130.00	130.00
3	2000	20	02335	December	2014	310.00	2000	101100	4	50.00	200.00
4	2000	20	02335	December	2014	310.00	2000	101200	2	50.00	100.00

SQL Outer Joins Example Tables

STUDENT

StudentPK	StudentName	LockerFK
1	Adams	NULL
2	Buchanan	NULL
3	Carter	10
4	Ford	20
5	Hoover	30
6	Kennedy	40
7	Roosevelt	50
8	Truman	60

LOCKER

LockerPK	LockerType
10	Full
20	Full
30	Half
40	Full
50	Full
60	Half
70	Full
80	Full
90	Half

The STUDENT and LOCKER Tables Aligned to Show Row Relationships

See Page 132(131~133)

SQL INNER JOIN

StudentPK	StudentName	LockerFK	LockerPK	LockerType
3	Carter	10	10	Full
4	Ford	20	20	Full
5	Hoover	30	30	Half
6	Kennedy	40	40	Full
7	Roosevelt	50	50	Full
8	Truman	60	60	Half

INNER JOIN of the STUDENT and LOCKER Tables

SQL LEFT OUTER JOIN

StudentPK	StudentName	LockerFK	LockerPK	LockerType
1	Adams	NULL	NULL	NULL
2	Buchanan	NULL	NULL	NULL
3	Carter	10	10	Full
4	Ford	20	20	Full
5	Hoover	30	30	Half
6	Kennedy	40	40	Full
7	Roosevelt	50	50	Full
8	Truman	60	60	Half

LEFT OUTER JOIN of the STUDENT and LOCKER Tables

SQL RIGHT OUTER JOIN

StudentPK	StudentName	LockerFK	LockerPK	LockerType
3	Carter	10	10	Full
4	Ford	20	20	Full
5	Hoover	30	30	Half
6	Kennedy	40	40	Full
7	Roosevelt	50	50	Full
8	Truman	60	60	Half
NULL	NULL	NULL	70	Full
NULL	NULL	NULL	80	Full
NULL	NULL	NULL	90	Half

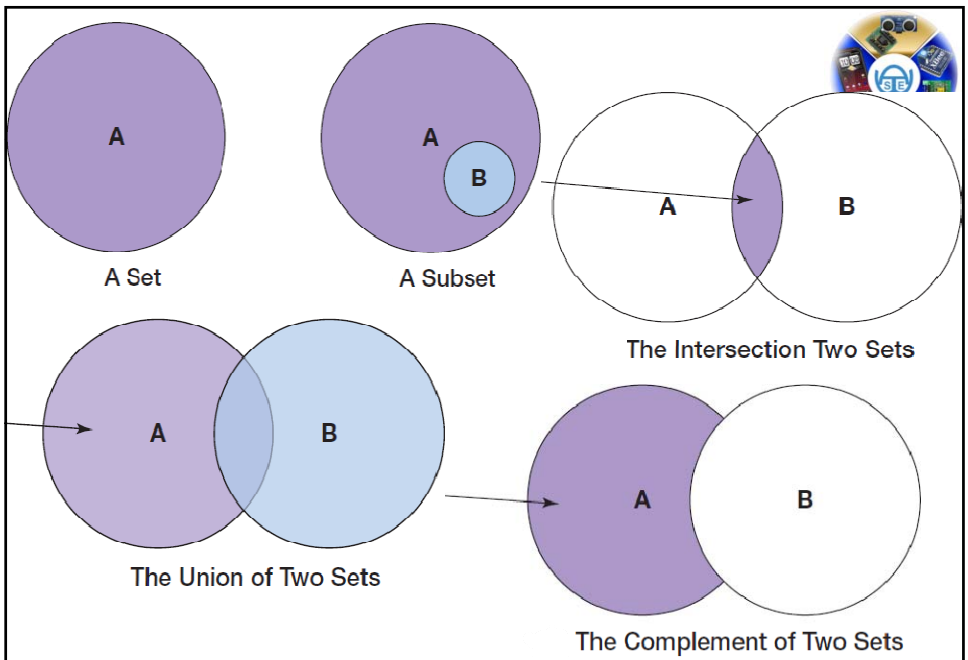
RIGHT OUTER JOIN of the STUDENT and LOCKER Tables

Querying Multiple Tables with SQL Set Operators

- Mathematicians use the term **set theory** to describe mathematical operations on sets, where a **set** is defined as a group of distinct items.
- A relational database table meets the definition of a set, so it is little wonder that SQL includes a group of **set operators** for use with SQL queries.

SQL Set Operators

- A **set** is represented by a labeled circle.
- A **subset** is a portion of a set that is contained entirely within the set.
- The **union** of two sets represents a set that contains all values in both sets. This is equivalent to an **OR** logical operation (**A OR B**).
- The **intersection** of two sets represents the area common to both sets. This is equivalent to an **AND** logical operation (**A AND B**).
- The **complement** of set B in set A represents everything in set A that is not in set B. This is equivalent to a logical operation using **NOT** (**A NOT B**).



Note that in order to use **SQL set operators**, the table columns involved in the operations ***must*** be the same number in each SELECT component, and corresponding columns ***must*** have the same or compatible (e.g., CHAR and VARCHAR) data types!

SQL Set Operators	
Operator	**Meaning**
UNION	The result is all the row values in one or both tables
INTERSECT	The result is all the row values common to both tables
EXCEPT	The result is all the row values in the first table but not the second

Querying Multiple Tables with SQL Set Operators

The Logic of Set Operators VIII – SQL UNION Operator

"What products were available for sale (by either catalog or Web site) in 2014 and 2015?"

```
/* *** SQL-Query-CH02-76 *** */
SELECT     SKU, SKU_Description, Department
FROM       CATALOG_SKU_2014
UNION
SELECT     SKU, SKU_Description, Department
FROM       CATALOG_SKU_2015;

/* *** SQL-Query-CH02-76-ALL *** */
SELECT     SKU, SKU_Description, Department
FROM       CATALOG_SKU_2014
UNION ALL
SELECT     SKU, SKU_Description, Department
FROM       CATALOG_SKU_2015;
```

See Page 136

SQL INTERSECT & EXCEPT Operator

```
/* *** SQL-Query-CH02-77 *** */
SELECT      SKU, SKU_Description, Department
FROM        CATALOG_SKU_2014
INTERSECT
SELECT      SKU, SKU_Description, Department
FROM        CATALOG_SKU_2015;
```

	SKU	SKU_Description	Department
1	100100	Std. Scuba Tank, Yellow	Water Sports
2	101100	Dive Mask, Small Clear	Water Sports
3	101200	Dive Mask, Med Clear	Water Sports
4	201000	Half-dome Tent	Camping
5	202000	Half-dome Tent Vestibule	Camping
6	301000	Light Fly Climbing Harness	Climbing
7	302000	Locking Carabiner, Oval	Climbing

```
/* *** SQL-Query-CH02-78 *** */
SELECT      SKU, SKU_Description, Department
FROM        CATALOG_SKU_2014
EXCEPT
SELECT      SKU, SKU_Description, Department
FROM        CATALOG_SKU_2015;
```

	SKU	SKU_Description	Department
1	100300	Std. Scuba Tank, Light Blue	Water Sports
2	100400	Std. Scuba Tank, Dark Blue	Water Sports

Homework

- 2.34 – 2.60