



# DAVID M. KROENKE and DAVID J. AUER DATABASE CONCEPTS, 4<sup>th</sup> Edition

## Structured Query Language

Course: DE4, Course Teacher: Dr. D. M. Akbar Hussain  
Department of Electronic Systems



## Chapter Objectives

- Learn basic SQL statements for creating database structures
- Learn SQL statements to add data to a database
- Learn basic SQL SELECT statements and options for processing a single table
- Learn basic SQL SELECT statements for processing multiple tables with sub-queries
- Learn basic SQL SELECT statements for processing multiple tables with joins
- Learn SQL statements to modify and delete data from a database
- Learn SQL statements to modify and delete database tables and constraints

Course: DE4, Course Teacher: Dr. D. M. Akbar Hussain  
Department of Electronic Systems

3-2



## Structured Query Language

- Structured Query Language
  - Acronym: SQL
  - Pronounced as "S-Q-L" ["Ess-Que-El"]
  - Originally developed by IBM as the SEQUEL language in the 1970s
  - SQL-92 is an ANSI national standard adopted in 1992
  - SQL:2008 is current standard



## SQL Defined

- SQL is not a programming language, but rather a data sub-language, SQL is comprised of
  - A Data Definition Language (DDL)
    - ✓ Used to define database structures
  - A Data Manipulation Language (DML)
    - ✓ Data definition and updating
    - ✓ Data retrieval (Queries)
  - There are other SQL functions
    - ✓ Concurrency control
    - ✓ Transaction control



## SQL for Data Definition

- The SQL Data Definition Statements include
  - **CREATE**
    - ✓ To create database objects
  - **ALTER**
    - ✓ To modify the structure and/or characteristics of database objects
  - **DROP**
    - ✓ To delete database objects



## SQL for Data Definition: **CREATE**

- Creating database tables
  - The SQL CREATE TABLE statement

```
CREATE TABLE EMPLOYEE (  
    EmpID      Integer      PRIMARY KEY,  
    EmpName    Char(25)     NOT NULL  
);
```



## SQL for Data Definition: CREATE with CONSTRAINT I

- Creating database tables with PRIMARY KEY constraints:
  - The SQL CREATE TABLE statement
  - The SQL CONSTRAINT keyword

```
CREATE TABLE EMPLOYEE(
    EmpID      Integer      NOT NULL,
    EmpName    Char(25)    NOT NULL,
    CONSTRAINT Emp_PK     PRIMARY KEY(EmpID)
);
```



## SQL for Data Definition: CREATE with CONSTRAINT II

- Creating database tables with composite primary keys using PRIMARY KEY constraints
  - The SQL CREATE TABLE statement
  - The SQL CONSTRAINT keyword

```
CREATE TABLE EMP_SKILL(
    EmpID      Integer      NOT NULL,
    SkillID    Integer      NOT NULL,
    SkillLevel Integer      NULL,
    CONSTRAINT EmpSkill_PK PRIMARY KEY
                        (EmpID, SkillID)
);
```

## SQL for Data Definition: CREATE with CONSTRAINT III

- Creating database tables using PRIMARY KEY and FOREIGN KEY constraints

- The SQL CREATE TABLE statement
- The SQL CONSTRAINT keyword

```
CREATE TABLE EMP_SKILL(
    EmpID      Integer      NOT NULL,
    SkillID    Integer      NOT NULL,
    SkillLevel Integer      NULL,
    CONSTRAINT EmpSkill_PK PRIMARY KEY
                        (EmpID, SkillID),
    CONSTRAINT Emp_FK    FOREIGN KEY (EmpID)
                        REFERENCES EMPLOYEE (EmpID),
    CONSTRAINT Skill_FK FOREIGN KEY (SkillID)
                        REFERENCES SKILL (SkillID)
);
```

## SQL for Data Definition: CREATE with CONSTRAINT III

```
CREATE TABLE DEPARTMENT(
    DepartmentName Char (35)      NOT NULL,
    BudgetCode     Char (15)      NOT NULL,
    OfficeNumber   Char (25)      NOT NULL,
    Phone          Char (12)      NOT NULL,
    CONSTRAINT DEPARTMENT_PK PRIMARY KEY (DepartmentName)
);

CREATE TABLE EMPLOYEE(
    EmployeeNumber Int           NOT NULL,
    FirstName       Char (25)    NOT NULL,
    LastName        Char (25)    NOT NULL,
    Department      Char (35)    NOT NULL,
    Phone          Char (12)     NULL,
    Email          Char(100)     NOT NULL,
    CONSTRAINT EMPLOYEE_PK PRIMARY KEY (EmployeeNumber),
    CONSTRAINT EMP_DEPART_FK FOREIGN KEY (Department)
                        REFERENCES DEPARTMENT (DepartmentName)
);
```



## SQL for Data Definition: CREATE with CONSTRAINT IV

- Creating database tables using PRIMARY KEY and FOREIGN KEY constraints

- The SQL CREATE TABLE statement
- The SQL CONSTRAINT keyword
- ON UPDATE CASCADE and ON DELETE CASCADE

```
CREATE TABLE EMP_SKILL(
  EmpID      Integer      NOT NULL,
  SkillID    Integer      NOT NULL,
  SkillLevel Integer      NULL,
  CONSTRAINT EmpSkill_PK PRIMARY KEY(EmpID, SkillID),
  CONSTRAINT Emp_FK FOREIGN KEY(EmpID)
    REFERENCES EMPLOYEE(EmpID)
    ON DELETE CASCADE,
  CONSTRAINT Skill_FK FOREIGN KEY(SkillID)
    REFERENCES SKILL(SkillID)
    ON UPDATE CASCADE
);
```

Course: DE4, Course Teacher: Dr. D. M. Akbar Hussain  
Department of Electronic Systems

3-11



## Primary Key Constraint: ALTER I

- Adding primary key constraints to an existing table

- ✓ The SQL ALTER statement

```
ALTER TABLE EMPLOYEE
  ADD CONSTRAINT Emp_PK PRIMARY KEY(EmpID);
```

Course: DE4, Course Teacher: Dr. D. M. Akbar Hussain  
Department of Electronic Systems

3-12



## Modifying Data using SQL: Removing a Constraint: ALTER & DROP

- To change the constraints on existing tables, you may need to remove the existing constraints before adding new constraints

```
ALTER TABLE EMPLOYEE DROP CONSTRAINT EmpFK;
```



## Composite Primary Key Constraints: ALTER II

- Adding a composite primary key constraint to an existing table:

➤ The SQL ALTER statement

```
ALTER TABLE EMP_SKILL  
ADD CONSTRAINT EmpSkill_PK  
PRIMARY KEY(EmpID, SkillID);
```



## Foreign Key Constraint: ALTER III

- Adding foreign key constraints to an existing table:

➤ The SQL ALTER statement

```
ALTER TABLE EMPLOYEE ADD  
CONSTRAINT Emp_FK FOREIGN KEY (DeptID)  
REFERENCES DEPARTMENT (DeptID);
```



## Modifying Data using SQL

- Insert
  - Will add a new row in a table (already discussed above)
- Update
  - Will update the data in a table that matches the specified criteria
- Delete
  - Will delete the data in a table that matches the specified criteria





## Adding Data: INSERT

- To add a row to an existing table, use the INSERT statement
- Non-numeric data must be enclosed in straight ( ' ) quotes

```
INSERT INTO EMPLOYEE (EmpID, SalaryCode)
VALUES {62, 11};
```



## Modifying Data using SQL: Changing Data Values: UPDATE

- To change the data values in an existing row (or set of rows) use the Update statement

```
UPDATE    EMPLOYEE
SET       Phone '791-555-1234'
WHERE     EmpID = 29;
```



## Modifying Data using SQL: Deleting Data: DELETE

- To delete a row or set of rows from a table using the DELETE statement

```
DELETE FROM EMPLOYEE  
WHERE EmpID = 29;
```



## Modifying Data using SQL: Deleting Database Objects: DROP

- To remove unwanted database objects from the database, use the SQL DROP statement
- Warning... The DROP statement will permanently remove the object and all data

```
DROP TABLE EMPLOYEE;
```



## Modifying Data using SQL:

The UPDATE command can modify more than one column value at a time, as shown in the following statement. For example, if Heather Jones (EmployeeNumber = 5) is transferred to the finance department from accounting and given a new finance phone number, you can use the following command to update her data:

```
UPDATE EMPLOYEE
SET Department = 'Finance', Phone = '360-285-8420',
WHERE EmployeeNumber = 5;
```

This command changes the values of Phone and Department for the indicated employee. You can use a SELECT command to see the results:

```
SELECT * FROM EMPLOYEE
WHERE EmployeeNumber = 5;
```

The results are as follows:

EmployeeNumber	FirstName	LastName	Department	Phone	Email	
1	5	Heather	Jones	Finance	360-285-8420	HJones@WPC.com



## Modifying Data using SQL: The CHECK Constraint

- The CHECK constraint can be used to create sets of values that restrict the values that can be used in a column

```
ALTER TABLE PROJECT
ADD CONSTRAINT PROJECT_Check_Dates
CHECK (StartDate < EndDate);
```

## SQL for Data Retrieval: Queries

- `SELECT` is the best known SQL statement
- `SELECT` will retrieve information from the database that matches the specified criteria using the `SELECT/FROM/WHERE` framework

```
SELECT EmpName
FROM EMPLOYEE
WHERE EmpID = 2007001;
```

## SQL for Data Retrieval: Queries

ProjectID	Name	Department	MaxHours	StartDate	EndDate	Add New Field
1000	2008 Q3 Product Plan	Marketing	135,00	10-05-2008	15-06-2008	
1100	2008 Q3 Portfolio Analysis	Finance	120,00	05-07-2008	25-07-2008	
1200	2008 Q3 Tax Preparation	Accounting	145,00	10-08-2008	15-10-2008	
1300	2008 Q4 Product Plan	Marketing	150,00	10-08-2008	15-09-2008	
1400	2008 Q4 Portfolio Analysis	Finance	140,00	05-10-2008		

### Reading Specified Columns from a Single Table

```
SELECT Name, Department, MaxHours
FROM PROJECT;
```



## SQL for Data Retrieval: Queries

ProjectID	Name	Department	MaxHours	StartDate	EndDate	Add New Field
1000	2008 Q3 Product Plan	Marketing	135,00	10-05-2008	15-06-2008	
1100	2008 Q3 Portfolio Analysis	Finance	120,00	05-07-2008	25-07-2008	
1200	2008 Q3 Tax Preparation	Accounting	145,00	10-08-2008	15-10-2008	
1300	2008 Q4 Product Plan	Marketing	150,00	10-08-2008	15-09-2008	
1400	2008 Q4 Portfolio Analysis	Finance	140,00	05-10-2008		

```
SELECT Name, MaxHours, Department
FROM PROJECT;
```



## SQL for Data Retrieval: Queries

ProjectID	Name	Department	MaxHours	StartDate	EndDate	Add New Field
1000	2008 Q3 Product Plan	Marketing	135,00	10-05-2008	15-06-2008	
1100	2008 Q3 Portfolio Analysis	Finance	120,00	05-07-2008	25-07-2008	
1200	2008 Q3 Tax Preparation	Accounting	145,00	10-08-2008	15-10-2008	
1300	2008 Q4 Product Plan	Marketing	150,00	10-08-2008	15-09-2008	
1400	2008 Q4 Portfolio Analysis	Finance	140,00	05-10-2008		

```
SELECT ProjectID, Name, Department, MaxHours,
       StartDate, EndDate
FROM PROJECT
WHERE Department = 'Finance';
```



## SQL for Data Retrieval: The Results of a Query is a Relation

- A query pulls information from one or more relations and creates (temporarily) a new relation, this allows a query to:
  - Create a new relation
  - Feed information to another query (as a "sub-query")



## SQL for Data Retrieval: Displaying All Columns

- To show all of the column values for the rows that match the specified criteria, use an asterisk ( \* )

```
SELECT *
FROM PROJECT;
```

ProjectID	Name	Department	MaxHours	StartDate	EndDate	Add New Field
1000	2008 Q3 Product Plan	Marketing	135,00	10-05-2008	15-06-2008	
1100	2008 Q3 Portfolio Analysis	Finance	120,00	05-07-2008	25-07-2008	
1200	2008 Q3 Tax Preparation	Accounting	145,00	10-08-2008	15-10-2008	
1300	2008 Q4 Product Plan	Marketing	150,00	10-08-2008	15-09-2008	
1400	2008 Q4 Portfolio Analysis	Finance	140,00	05-10-2008		

## SQL for Data Retrieval: Showing Each Row Only Once

ProjectID	Name	Department	MaxHours	StartDate	EndDate	Add New Field
1000	2008 Q3 Product Plan	Marketing	135,00	10-05-2008	15-06-2008	
1100	2008 Q3 Portfolio Analysis	Finance	120,00	05-07-2008	25-07-2008	
1200	2008 Q3 Tax Preparation	Accounting	145,00	10-08-2008	15-10-2008	
1300	2008 Q4 Product Plan	Marketing	150,00	10-08-2008	15-09-2008	
1400	2008 Q4 Portfolio Analysis	Finance	140,00	05-10-2008		

- The **DISTINCT** keyword may be added to the **SELECT** statement to inhibit duplicate rows from displaying

```
SELECT DISTINCT Department
FROM PROJECT;
```

Department
Marketing
Finance
Accounting

Department
Marketing
Finance
Accounting
Marketing
Finance

## SQL for Data Retrieval: Specifying Search Criteria

ProjectID	Name	Department	MaxHours	StartDate	EndDate	Add New Field
1000	2008 Q3 Product Plan	Marketing	135,00	10-05-2008	15-06-2008	
1100	2008 Q3 Portfolio Analysis	Finance	120,00	05-07-2008	25-07-2008	
1200	2008 Q3 Tax Preparation	Accounting	145,00	10-08-2008	15-10-2008	
1300	2008 Q4 Product Plan	Marketing	150,00	10-08-2008	15-09-2008	
1400	2008 Q4 Portfolio Analysis	Finance	140,00	05-10-2008		

- The **WHERE** clause stipulates the matching criteria for the record that are to be displayed

```
SELECT      Name
FROM        PROJECT
WHERE       MaxHours = 120.00;
```



## SQL for Data Retrieval: Match Criteria

- The WHERE clause match criteria may include:

- Equals "="
- Not Equals "<>"
- Greater than ">"
- Less than "<"
- Greater than or Equal to ">="
- Less than or Equal to "<="



## SQL for Data Retrieval: Match Criteria

ProjectID	Name	Department	MaxHours	StartDate	EndDate	Add New Field
1000	2008 Q3 Product Plan	Marketing	135,00	10-05-2008	15-06-2008	
1100	2008 Q3 Portfolio Analysis	Finance	120,00	05-07-2008	25-07-2008	
1200	2008 Q3 Tax Preparation	Accounting	145,00	10-08-2008	15-10-2008	
1300	2008 Q4 Product Plan	Marketing	150,00	10-08-2008	15-09-2008	
1400	2008 Q4 Portfolio Analysis	Finance	140,00	05-10-2008		

```
SELECT *
FROM PROJECT
WHERE MaxHours > 135;
```





## SQL for Data Retrieval: Match Operators

- Multiple matching criteria may be specified using:
  - **AND**
    - ✓ Representing an intersection of the data sets
  - **OR**
    - ✓ Representing a union of the data sets



## SQL for Data Retrieval: Operator Examples

```
SELECT    EmpName
FROM      EMPLOYEE
WHERE     DeptID < 7
         OR   DeptID > 12;
```

```
SELECT    EmpName
FROM      EMPLOYEE
WHERE     DeptID = 9
         AND SalaryCode <= 23;
```



## SQL for Data Retrieval: Operator Examples

ProjectID	Name	Department	MaxHours	StartDate	EndDate	Add New Field
1000	2008 Q3 Product Plan	Marketing	135,00	10-05-2008	15-06-2008	
1100	2008 Q3 Portfolio Analysis	Finance	120,00	05-07-2008	25-07-2008	
1200	2008 Q3 Tax Preparation	Accounting	145,00	10-08-2008	15-10-2008	
1300	2008 Q4 Product Plan	Marketing	150,00	10-08-2008	15-09-2008	
1400	2008 Q4 Portfolio Analysis	Finance	140,00	05-10-2008		

```

SELECT *
FROM PROJECT
WHERE Department = 'Finance' AND MaxHours > 135;
    
```



## SQL for Data Retrieval: Operator Examples

EmployeeNumb.	FirstName	LastName	Department	Phone	Email
1	Mary	Jacobs	Administration	360-285-8110	MJacobs@WPC.com
2	Rosalie	Jackson	Administration	360-285-8120	RJackson@WPC.com
3	Richard	Banalone	Legal	360-285-8210	RBanalone@WPC.com
4	Tom	Caruthers	Accounting	360-285-8310	TCaruthers@WPC.com
5	Heather	Jones	Finance	360-285-8420	HJones@WPC.com
6	Mary	Abernathy	Finance	360-285-8410	MAbernathy@WPC.com
7	George	Smith	Human Resources	360-285-8510	GSmith@WPC.com
8	Tom	Jackson	Production	360-287-8610	TJackson@WPC.com
9	George	Jones	Production	360-287-8620	GJones@WPC.com
10	Ken	Numoto	Marketing	360-287-8710	KMumoto@WPC.com
11	James	Nestor	Info Systems	360-287-8710	JNestor@WPC.com
12	Rick	Brown	Info Systems	360-287-8820	RBrown@WPC.com

```

SELECT FirstName, LastName, Phone, Department
FROM EMPLOYEE
WHERE Department = 'Accounting'
AND Phone = '360-285-8310';
    
```



## SQL for Data Retrieval: Operator Examples

EmployeeNum	FirstName	LastName	Department	Phone	Email
1	Mary	Jacobs	Administration	360-285-8110	MJacobs@WPC.com
2	Rosalie	Jackson	Administration	360-285-8120	RJackson@WPC.com
3	Richard	Banalone	Legal	360-285-8210	RBanalone@WPC.com
4	Tom	Caruthers	Accounting	360-285-8310	TCaruthers@WPC.com
5	Heather	Jones	Finance	360-285-8420	HJones@WPC.com
6	Mary	Abermathy	Finance	360-285-8410	MAbermathy@WPC.com
7	George	Smith	Human Resources	360-285-8510	GSmith@WPC.com
8	Tom	Jackson	Production	360-287-8610	TJackson@WPC.com
9	George	Jones	Production	360-287-8620	GJones@WPC.com
10	Ken	Numoto	Marketing	360-287-8710	KNumoto@WPC.com
11	James	Nestor	Info Systems	360-287-8810	JNestor@WPC.com
12	Rick	Brown	Info Systems	360-287-8820	RBrown@WPC.com

```

SELECT  FirstName, LastName, Phone, Department
FROM    EMPLOYEE
WHERE   Department = 'Accounting'
        OR Phone = '360-285-8410';

```



## SQL for Data Retrieval: A List of Values

- The WHERE clause may include the IN keyword to specify that a particular column value must be included in a list of values

```

SELECT  EmpName
FROM    EMPLOYEE
WHERE   DeptID IN (4, 8, 9);

```



## SQL for Data Retrieval: A List of Values

EmployeeNum	FirstName	LastName	Department	Phone	Email
1	Mary	Jacobs	Administration	360-285-8110	Mjacobs@WPC.com
2	Rosalie	Jackson	Administration	360-285-8120	RJackson@WPC.com
3	Richard	Bandalone	Legal	360-285-8210	RBandalone@WPC.com
4	Tom	Caruthers	Accounting	360-285-8310	TCaruthers@WPC.com
5	Heather	Jones	Finance	360-285-8420	HJones@WPC.com
6	Mary	Abernathy	Finance	360-285-8410	MAbernathy@WPC.com
7	George	Smith	Human Resources	360-285-8510	GSmith@WPC.com
8	Tom	Jackson	Production	360-287-8610	TJackson@WPC.com
9	George	Jones	Production	360-287-8620	GJones@WPC.com
10	Ken	Numoto	Marketing	360-287-8710	KNumoto@WPC.com
11	James	Nestor	Info Systems		JNestor@WPC.com
12	Rick	Brown	Info Systems	360-287-8820	RBrown@WPC.com

```
SELECT  FirstName, LastName, Phone, Department
FROM    EMPLOYEE
WHERE   Department IN ('Accounting', 'Finance',
                      'Marketing');
```

Course: DE4, Course Teacher: Dr. D. M. Akbar Hussain  
Department of Electronic Systems

3-39



## SQL for Data Retrieval: The Logical NOT Operator

- Any criteria statement may be preceded by a NOT operator which is to say that all information will be shown except that information matching the specified criteria

```
SELECT  EmpName
FROM    EMPLOYEE
WHERE   DeptID NOT IN (4, 8, 9);
```

Course: DE4, Course Teacher: Dr. D. M. Akbar Hussain  
Department of Electronic Systems

3-40

## SQL for Data Retrieval: A List of Values

EmployeeNum	FirstName	LastName	Department	Phone	Email
1	Mary	Jacobs	Administration	360-285-8110	MJacobs@WPC.com
2	Rosalie	Jackson	Administration	360-285-8120	RJackson@WPC.com
3	Richard	Banalone	Legal	360-285-8210	RBanalone@WPC.com
4	Tom	Caruthers	Accounting	360-285-8310	TCaruthers@WPC.com
5	Heather	Jones	Finance	360-285-8420	HJones@WPC.com
6	Mary	Abernathy	Finance	360-285-8410	MAbernathy@WPC.com
7	George	Smith	Human Resources	360-285-8510	GSmith@WPC.com
8	Tom	Jackson	Production	360-287-8610	TJackson@WPC.com
9	George	Jones	Production	360-287-8620	GJones@WPC.com
10	Ken	Numoto	Marketing	360-287-8710	KMumoto@WPC.com
11	James	Nestor	Info Systems		JNestor@WPC.com
12	Rick	Brown	Info Systems	360-287-8820	RBrown@WPC.com

```
SELECT  FirstName, LastName, Phone, Department
FROM    EMPLOYEE
WHERE   Department NOT IN ('Accounting', 'Finance',
                          'Marketing');
```

Course: DE4, Course Teacher: Dr. D. M. Akbar Hussain  
Department of Electronic Systems

3-41

## SQL for Data Retrieval: Finding Data in a Range of Values

- SQL provides a **BETWEEN** keyword that allows a user to specify a minimum and maximum value on one line

```
SELECT  EmpName
FROM    EMPLOYEE
WHERE   SalaryCode BETWEEN 10 AND 45;
```

Course: DE4, Course Teacher: Dr. D. M. Akbar Hussain  
Department of Electronic Systems

3-42



## SQL for Data Retrieval: A List of Values

EmployeeNumber	FirstName	LastName	Department	Phone	Email
1	Mary	Jacobs	Administration	360-285-8110	MJacobs@WPC.com
2	Rosalie	Jackson	Administration	360-285-8120	RJackson@WPC.com
3	Richard	Bandalone	Legal	360-285-8210	RBandalone@WPC.com
4	Tom	Caruthers	Accounting	360-285-8310	TCaruthers@WPC.com
5	Heather	Jones	Finance	360-285-8420	HJones@WPC.com
6	Mary	Abernathy	Finance	360-285-8410	MAbernathy@WPC.com
7	George	Smith	Human Resources	360-285-8510	GSmith@WPC.com
8	Tom	Jackson	Production	360-287-8610	TJackson@WPC.com
9	George	Jones	Production	360-287-8620	GJones@WPC.com
10	Ken	Numoto	Marketing	360-287-8710	KMumoto@WPC.com
11	James	Nestor	Info Systems		JNestor@WPC.com
12	Rick	Brown	Info Systems	360-287-8820	RBrown@WPC.com

```

SELECT  FirstName, LastName, Phone, Department
FROM    EMPLOYEE
WHERE   EmployeeNumber BETWEEN 2 AND 5;

```

Course: DE4, Course Teacher: Dr. D. M. Akbar Hussain  
Department of Electronic Systems

3-43



## SQL for Data Retrieval: Allowing for Wildcard Searches

- The SQL LIKE keyword allows searches on partial data values
- LIKE can be paired with wildcards to find rows matching a string value:
  - Multiple character wildcard character is a percent sign (%)
  - Single character wildcard character is an underscore (\_)

Course: DE4, Course Teacher: Dr. D. M. Akbar Hussain  
Department of Electronic Systems

3-44



## SQL for Data Retrieval: Wildcard Search Examples

```
SELECT EmpID
FROM EMPLOYEE
WHERE EmpName LIKE 'Kr%';
```

```
SELECT EmpID
FROM EMPLOYEE
WHERE Phone LIKE '616-____-____';
```



## SQL for Data Retrieval: Wildcard Search Examples

ProjectID	Name	Department	MaxHours	StartDate	EndDate	Add New Field
1000	2008 Q3 Product Plan	Marketing	135,00	10-05-2008	15-06-2008	
1100	2008 Q3 Portfolio Analysis	Finance	120,00	05-07-2008	25-07-2008	
1200	2008 Q3 Tax Preparation	Accounting	145,00	10-08-2008	15-10-2008	
1300	2008 Q4 Product Plan	Marketing	150,00	10-08-2008	15-09-2008	
1400	2008 Q4 Portfolio Analysis	Finance	140,00	05-10-2008		

```
SELECT *
FROM PROJECT
WHERE Name LIKE '2008 Q_ Portfolio Analysis';
```



## SQL for Data Retrieval: Wildcard Search Examples

EmployeeNum	FirstName	LastName	Department	Phone	Email
1	Mary	Jacobs	Administration	360-285-8110	MJacobs@WPC.com
2	Rosalie	Jackson	Administration	360-285-8120	RJackson@WPC.com
3	Richard	Bandalone	Legal	360-285-8210	RBandalone@WPC.com
4	Tom	Caruthers	Accounting	360-285-8310	TCaruthers@WPC.com
5	Heather	Jones	Finance	360-285-8420	HJones@WPC.com
6	Mary	Abernathy	Finance	360-285-8410	MAbernathy@WPC.com
7	George	Smith	Human Resources	360-285-8510	GSmith@WPC.com
8	Tom	Jackson	Production	360-287-8610	TJackson@WPC.com
9	George	Jones	Production	360-287-8620	GJones@WPC.com
10	Ken	Numoto	Marketing	360-287-8710	KMumoto@WPC.com
11	James	Nestor	Info Systems	360-287-8710	JNestor@WPC.com
12	Rick	Brown	Info Systems	360-287-8820	RBrown@WPC.com

```
SELECT *
FROM EMPLOYEE
WHERE Phone LIKE '360-287-____';
```

Course: DE4, Course Teacher: Dr. D. M. Akbar Hussain  
Department of Electronic Systems

3-47



## SQL for Data Retrieval: Wildcard Search Examples

EmployeeNum	FirstName	LastName	Department	Phone	Email
1	Mary	Jacobs	Administration	360-285-8110	MJacobs@WPC.com
2	Rosalie	Jackson	Administration	360-285-8120	RJackson@WPC.com
3	Richard	Bandalone	Legal	360-285-8210	RBandalone@WPC.com
4	Tom	Caruthers	Accounting	360-285-8310	TCaruthers@WPC.com
5	Heather	Jones	Finance	360-285-8420	HJones@WPC.com
6	Mary	Abernathy	Finance	360-285-8410	MAbernathy@WPC.com
7	George	Smith	Human Resources	360-285-8510	GSmith@WPC.com
8	Tom	Jackson	Production	360-287-8610	TJackson@WPC.com
9	George	Jones	Production	360-287-8620	GJones@WPC.com
10	Ken	Numoto	Marketing	360-287-8710	KMumoto@WPC.com
11	James	Nestor	Info Systems	360-287-8710	JNestor@WPC.com
12	Rick	Brown	Info Systems	360-287-8820	RBrown@WPC.com

```
SELECT *
FROM EMPLOYEE
WHERE Phone LIKE '360-287-%';
```

Course: DE4, Course Teacher: Dr. D. M. Akbar Hussain  
Department of Electronic Systems

3-48





## SQL for Data Retrieval: Wildcard Search Examples

EmployeeNum	FirstName	LastName	Department	Phone	Email
1	Mary	Jacobs	Administration	360-285-8110	MJacobs@WPC.com
2	Rosalie	Jackson	Administration	360-285-8120	RJackson@WPC.com
3	Richard	Banalone	Legal	360-285-8210	RBanalone@WPC.com
4	Tom	Caruthers	Accounting	360-285-8216	TCaruthers@WPC.com
5	Heather	Jones	Finance	360-285-8420	HJones@WPC.com
6	Mary	Abernathy	Finance	360-285-8410	MAbernathy@WPC.com
7	George	Smith	Human Resources	360-285-8510	GSmith@WPC.com
8	Tom	Jackson	Production	360-287-8610	TJackson@WPC.com
9	George	Jones	Production	360-287-8620	GJones@WPC.com
10	Ken	Numoto	Marketing	360-287-8710	KMumoto@WPC.com
11	James	Nestor	Info Systems		JNestor@WPC.com
12	Rick	Brown	Info Systems	360-287-8820	RBrown@WPC.com

```
SELECT *
FROM EMPLOYEE
WHERE Department LIKE '%ing';
```

Course: DE4, Course Teacher: Dr. D. M. Akbar Hussain  
Department of Electronic Systems

3-49



## SQL for Data Retrieval: Sorting the Results

- Query results may be sorted using the ORDER BY clause

```
SELECT *
FROM EMPLOYEE
ORDER BY EmpName;
```

Course: DE4, Course Teacher: Dr. D. M. Akbar Hussain  
Department of Electronic Systems

3-50



## SQL for Data Retrieval: Sorting the Results

EmployeeID	FirstName	LastName	Department	Phone	Email
1	Mary	Jacobs	Administration	360-285-8110	MJacobs@WPC.com
2	Rosalie	Jackson	Administration	360-285-8120	RJackson@WPC.com
3	Richard	Bandalone	Legal	360-285-8210	RBandalone@WPC.com
4	Tom	Caruthers	Accounting	360-285-8310	TCaruthers@WPC.com
5	Heather	Jones	Finance	360-285-8420	HJones@WPC.com
6	Mary	Abemathy	Finance	360-285-8410	MAbemathy@WPC.com
7	George	Smith	Human Resources	360-285-8510	GSmith@WPC.com
8	Tom	Jackson	Production	360-287-8610	TJackson@WPC.com
9	George	Jones	Production	360-287-8620	GJones@WPC.com
10	Ken	Numoto	Marketing	360-287-8710	KNumoto@WPC.com
11	James	Nestor	Info Systems	360-287-8810	JNestor@WPC.com
12	Rick	Brown	Info Systems	360-287-8820	RBrown@WPC.com

```
SELECT FirstName, LastName, Phone, Department
FROM EMPLOYEE
ORDER BY Department;
```

Course: DE4, Course Teacher: Dr. D. M. Akbar Hussain  
Department of Electronic Systems

3-51



## SQL for Data Retrieval: Sorting the Results

EmployeeID	FirstName	LastName	Department	Phone	Email
1	Mary	Jacobs	Administration	360-285-8110	MJacobs@WPC.com
2	Rosalie	Jackson	Administration	360-285-8120	RJackson@WPC.com
3	Richard	Bandalone	Legal	360-285-8210	RBandalone@WPC.com
4	Tom	Caruthers	Accounting	360-285-8310	TCaruthers@WPC.com
5	Heather	Jones	Finance	360-285-8420	HJones@WPC.com
6	Mary	Abemathy	Finance	360-285-8410	MAbemathy@WPC.com
7	George	Smith	Human Resources	360-285-8510	GSmith@WPC.com
8	Tom	Jackson	Production	360-287-8610	TJackson@WPC.com
9	George	Jones	Production	360-287-8620	GJones@WPC.com
10	Ken	Numoto	Marketing	360-287-8710	KNumoto@WPC.com
11	James	Nestor	Info Systems	360-287-8810	JNestor@WPC.com
12	Rick	Brown	Info Systems	360-287-8820	RBrown@WPC.com

```
SELECT FirstName, LastName, Phone, Department
FROM EMPLOYEE
ORDER BY Department DESC;
```

Course: DE4, Course Teacher: Dr. D. M. Akbar Hussain  
Department of Electronic Systems

3-52



## SQL for Data Retrieval: Sorting the Results

EmployeeNumber	FirstName	LastName	Department	Phone	Email
1	Mary	Jacobs	Administration	360-285-8110	MJacobs@WPC.com
2	Rosalie	Jackson	Administration	360-285-8120	RJackson@WPC.com
3	Richard	Bardalone	Legal	360-285-8210	RBardalone@WPC.com
4	Tom	Caruthers	Accounting	360-285-8310	TCaruthers@WPC.com
5	Heather	Jones	Finance	360-285-8420	HJones@WPC.com
6	Mary	Abernathy	Finance	360-285-8410	MAbernathy@WPC.com
7	George	Smith	Human Resources	360-285-8510	GSmith@WPC.com
8	Tom	Jackson	Production	360-287-8610	TJackson@WPC.com
9	George	Jones	Production	360-287-8620	GJones@WPC.com
10	Ken	Numoto	Marketing	360-287-8710	KNumoto@WPC.com
11	James	Nester	Info Systems		JNester@WPC.com
12	Rick	Brown	Info Systems	360-287-8820	RBrown@WPC.com

```
SELECT  FirstName, LastName, Phone, Department
FROM    EMPLOYEE
ORDER BY Department DESC, LastName ASC;
```

Course: DE4, Course Teacher: Dr. D. M. Akbar Hussain  
Department of Electronic Systems

3-53



## SQL for Data Retrieval: Built-in SQL Functions

- SQL provides several built-in functions
  - **COUNT**
    - ✓ Counts the number of rows that match the specified criteria
  - **MIN**
    - ✓ Finds the minimum value for a specific column for those rows matching the criteria
  - **MAX**
    - ✓ Finds the maximum value for a specific column for those rows matching the criteria

Course: DE4, Course Teacher: Dr. D. M. Akbar Hussain  
Department of Electronic Systems

3-54



## SQL for Data Retrieval: Built-in SQL Functions (continued)

- **SUM**
  - Calculates the sum for a specific column for those rows matching the criteria
- **AVG**
  - Calculates the numerical average of a specific column for those rows matching the criteria



## SQL for Data Retrieval: Built-in Function Examples

ProjectID	Name	Department	MaxHours	StartDate	EndDate	Add New Field
1000	2008 Q3 Product Plan	Marketing	135,00	10-05-2008	15-06-2008	
1100	2008 Q3 Portfolio Analysis	Finance	120,00	05-07-2008	25-07-2008	
1200	2008 Q3 Tax Preparation	Accounting	145,00	10-08-2008	15-10-2008	
1300	2008 Q4 Product Plan	Marketing	150,00	10-08-2008	15-09-2008	
1400	2008 Q4 Portfolio Analysis	Finance	140,00	05-10-2008		

```
SELECT    COUNT(*)
FROM      PROJECT;
```

```
SELECT    COUNT(*) AS NumberOfProjects
FROM      PROJECT;
```



## SQL for Data Retrieval: Built-in Function Examples

ProjectID	Name	Department	MaxHours	StartDate	EndDate	Add New Field
1000	2008 Q3 Product Plan	Marketing	135,00	10-05-2008	15-06-2008	
1100	2008 Q3 Portfolio Analysis	Finance	120,00	05-07-2008	25-07-2008	
1200	2008 Q3 Tax Preparation	Accounting	145,00	10-08-2008	15-10-2008	
1300	2008 Q4 Product Plan	Marketing	150,00	10-08-2008	15-09-2008	
1400	2008 Q4 Portfolio Analysis	Finance	140,00	05-10-2008		

```
SELECT      COUNT(Department) AS NumberOfDepartments
FROM        PROJECT;
```

```
SELECT      COUNT(DISTINCT Department) AS NumberOfDepartments
FROM        PROJECT;
```

Course: DE4, Course Teacher: Dr. D. M. Akbar Hussain  
Department of Electronic Systems

3-57



## SQL for Data Retrieval: Built-in Function Examples

ProjectID	Name	Department	MaxHours	StartDate	EndDate	Add New Field
1000	2008 Q3 Product Plan	Marketing	135,00	10-05-2008	15-06-2008	
1100	2008 Q3 Portfolio Analysis	Finance	120,00	05-07-2008	25-07-2008	
1200	2008 Q3 Tax Preparation	Accounting	145,00	10-08-2008	15-10-2008	
1300	2008 Q4 Product Plan	Marketing	150,00	10-08-2008	15-09-2008	
1400	2008 Q4 Portfolio Analysis	Finance	140,00	05-10-2008		

```
SELECT      MIN(MaxHours) AS MinimumMaxHours,
            MAX(MaxHours) AS MaximumMaxHours,
            SUM(MaxHours) AS TotalMaxHours
FROM        PROJECT
WHERE       ProjectID <= 1200;
```

Course: DE4, Course Teacher: Dr. D. M. Akbar Hussain  
Department of Electronic Systems

3-58



## SQL for Data Retrieval: Built-in Function Examples

ProjectID	Name	Department	MaxHours	StartDate	EndDate	Add New Field
1000	2008 Q3 Product Plan	Marketing	135,00	10-05-2008	15-06-2008	
1100	2008 Q3 Portfolio Analysis	Finance	120,00	05-07-2008	25-07-2008	
1200	2008 Q3 Tax Preparation	Accounting	145,00	10-08-2008	15-10-2008	
1300	2008 Q4 Product Plan	Marketing	150,00	10-08-2008	15-09-2008	
1400	2008 Q4 Portfolio Analysis	Finance	140,00	05-10-2008		

```
SELECT ProjectID, Name, MaxHours,
       (18.50 * MaxHours) AS MaxProjectCost
FROM PROJECT;
```

Course: DE4, Course Teacher: Dr. D. M. Akbar Hussain  
Department of Electronic Systems

3-59



## SQL for Data Retrieval: Providing Subtotals: GROUP BY

- In SQL GROUP BY clause group rows of common values. This increases the utility of built in functions because one can apply them to group of rows, following statement counts the number of employees in each department.
- Subtotals may be calculated by using the GROUP BY clause
- The HAVING clause may be used to restrict which data is displayed

```
SELECT DeptID,
       COUNT(*) AS NumOfEmployees
FROM EMPLOYEE
GROUP BY DeptID
HAVING COUNT(*) > 3;
```

Course: DE4, Course Teacher: Dr. D. M. Akbar Hussain  
Department of Electronic Systems

3-60



## SQL for Data Retrieval: Providing Subtotals: GROUP BY

EmployeeNum	FirstName	LastName	Department	Phone	Email
1	Mary	Jacobs	Administration	360-285-8110	MJacobs@WPC.com
2	Rosalie	Jackson	Administration	360-285-8120	RJackson@WPC.com
3	Richard	Bandalone	Legal	360-285-8210	RBanalone@WPC.com
4	Tom	Caruthers	Accounting	360-285-8310	TCaruthers@WPC.com
5	Heather	Jones	Finance	360-285-8420	HJones@WPC.com
6	Mary	Abernathy	Finance	360-285-8410	MAbernathy@WPC.com
7	George	Smith	Human Resources	360-285-8510	GSmith@WPC.com
8	Tom	Jackson	Production	360-287-8610	TJackson@WPC.com
9	George	Jones	Production	360-287-8620	GJones@WPC.com
10	Ken	Numoto	Marketing	360-287-8710	KMumoto@WPC.com
11	James	Nestor	Info Systems	360-287-8710	JNestor@WPC.com
12	Rick	Brown	Info Systems	360-287-8820	RBrown@WPC.com

```
SELECT Department, Count(*) AS NumberOfEmployees
FROM EMPLOYEE
GROUP BY Department;
```

Course: DE4, Course Teacher: Dr. D. M. Akbar Hussain  
Department of Electronic Systems

3-61



## SQL for Data Retrieval: Providing Subtotals: GROUP BY

EmployeeNum	FirstName	LastName	Department	Phone	Email
1	Mary	Jacobs	Administration	360-285-8110	MJacobs@WPC.com
2	Rosalie	Jackson	Administration	360-285-8120	RJackson@WPC.com
3	Richard	Bandalone	Legal	360-285-8210	RBanalone@WPC.com
4	Tom	Caruthers	Accounting	360-285-8310	TCaruthers@WPC.com
5	Heather	Jones	Finance	360-285-8420	HJones@WPC.com
6	Mary	Abernathy	Finance	360-285-8410	MAbernathy@WPC.com
7	George	Smith	Human Resources	360-285-8510	GSmith@WPC.com
8	Tom	Jackson	Production	360-287-8610	TJackson@WPC.com
9	George	Jones	Production	360-287-8620	GJones@WPC.com
10	Ken	Numoto	Marketing	360-287-8710	KMumoto@WPC.com
11	James	Nestor	Info Systems	360-287-8710	JNestor@WPC.com
12	Rick	Brown	Info Systems	360-287-8820	RBrown@WPC.com

```
SELECT Department, Count(*) AS NumberOfEmployees
FROM EMPLOYEE
GROUP BY Department
HAVING COUNT(*) > 1;
```

Course: DE4, Course Teacher: Dr. D. M. Akbar Hussain  
Department of Electronic Systems

3-62



## SQL for Data Retrieval: Providing Subtotals: GROUP BY

EmployeeNumber	FirstName	LastName	Department	Phone	Email
1	Mary	Jacobs	Administration	360-285-8110	MJacobs@WPC.com
2	Rosalie	Jackson	Administration	360-285-8120	RJackson@WPC.com
3	Richard	Bandalone	Legal	360-285-8210	RBandalone@WPC.com
4	Tom	Caruthers	Accounting	360-285-8310	TCaruthers@WPC.com
5	Heather	Jones	Finance	360-285-8420	HJones@WPC.com
6	Mary	Abernathy	Finance	360-285-8410	MAbernathy@WPC.com
7	George	Smith	Human Resources	360-285-8510	GSmith@WPC.com
8	Tom	Jackson	Production	360-287-8610	TJackson@WPC.com
9	George	Jones	Production	360-287-8620	GJones@WPC.com
10	Ken	Numoto	Marketing	360-287-8710	KNumoto@WPC.com
11	James	Nestor	Info Systems		JNestor@WPC.com
12	Rick	Brown	Info Systems	360-287-8820	RBrown@WPC.com

```

SELECT Department, Count(*) AS NumberOfEmployees
FROM EMPLOYEE
WHERE EmployeeNumber <= 6
GROUP BY Department
HAVING COUNT(*) > 1;

```

Course: DE4, Course Teacher: Dr. D. M. Akbar Hussain  
Department of Electronic Systems

3-63



## SQL for Data Retrieval: Retrieving Information from Multiple Tables

- Sub-queries

- As stated earlier, the result of a query is a relation. As a result, a query may feed another query. This is called a *sub-query*.

- Joins

- Another way of combining data is by using a *join*
    - ✓ Join [also called an Inner Join]
    - ✓ Left Outer Join
    - ✓ Right Outer Join

Course: DE4, Course Teacher: Dr. D. M. Akbar Hussain  
Department of Electronic Systems

3-64





## SQL for Data Retrieval: Retrieving Information from Multiple Tables

ProjectID	EmployeeNumber	HoursWorked
1000	1	30,00
1000	8	75,00
1000	10	55,00
1100	4	40,00
1100	6	45,00
1200	1	25,00
1200	2	20,00
1200	4	45,00
1200	5	40,00
1300	1	35,00
1300	8	80,00
1300	10	50,00
1400	4	15,00
1400	5	10,00
1400	6	27,50

EmployeeNumber	FirstName	LastName	Department	Phone	Email
1	Mary	Jacobs	Administration	360-285-8110	MJacobs@WPC.com
2	Rosalie	Jackson	Administration	360-285-8120	RJackson@WPC.com
3	Richard	Bardalone	Legal	360-285-8210	RBardalone@WPC.com
4	Tom	Caruthers	Accounting	360-285-8310	TCaruthers@WPC.com
5	Heather	Jones	Finance	360-285-8420	HJones@WPC.com
6	Mary	Abernathy	Finance	360-285-8410	MAbernathy@WPC.com
7	George	Smith	Human Resources	360-285-8310	GSmith@WPC.com
8	Tom	Jackson	Production	360-287-8610	TJackson@WPC.com
9	George	Jones	Production	360-287-8620	GJones@WPC.com
10	Ken	Numoto	Marketing	360-287-8710	KNumoto@WPC.com
11	James	Nestor	Info Systems	360-287-8820	JNestor@WPC.com
12	Rick	Brown	Info Systems	360-287-8820	RBrown@WPC.com

The queries considered so far have involved data from a single table. However, at times, more than one table must be processed to obtain the desired information. For example, suppose we want to know the names of all employees who have worked more than 40 hours on any single assignment. The names of employees are stored in the EMPLOYEE table, but the hours they have worked are stored in the ASSIGNMENT table.



## SQL for Data Retrieval: Subquery Example

EmployeeNumber	FirstName	LastName	Department	Phone	Email
1	Mary	Jacobs	Administration	360-285-8110	MJacobs@WPC.com
2	Rosalie	Jackson	Administration	360-285-8120	RJackson@WPC.com
3	Richard	Bardalone	Legal	360-285-8210	RBardalone@WPC.com
4	Tom	Caruthers	Accounting	360-285-8310	TCaruthers@WPC.com
5	Heather	Jones	Finance	360-285-8420	HJones@WPC.com
6	Mary	Abernathy	Finance	360-285-8410	MAbernathy@WPC.com
7	George	Smith	Human Resources	360-285-8310	GSmith@WPC.com
8	Tom	Jackson	Production	360-287-8610	TJackson@WPC.com
9	George	Jones	Production	360-287-8620	GJones@WPC.com
10	Ken	Numoto	Marketing	360-287-8710	KNumoto@WPC.com
11	James	Nestor	Info Systems	360-287-8820	JNestor@WPC.com
12	Rick	Brown	Info Systems	360-287-8820	RBrown@WPC.com

ProjectID	EmployeeNumber	HoursWorked
1000	1	30,00
1000	8	75,00
1000	10	55,00
1100	4	40,00
1100	6	45,00
1200	1	25,00
1200	2	20,00
1200	4	45,00
1200	5	40,00
1300	1	35,00
1300	8	80,00
1300	10	50,00
1400	4	15,00
1400	5	10,00
1400	6	27,50

If we knew that employees with EmployeeNumber 8 and 10 have worked more than 50 hours on an assignment (which is true), we could obtain their names with the following expression:

```
SELECT FirstName, LastName
FROM EMPLOYEE
WHERE EmployeeNumber IN (8, 10);
```

The result is:

FirstName	LastName
1	Tom Jackson
2	Ken Numoto

But according to the problem description, we are not given the employee numbers. We can, however, obtain the appropriate employee numbers with the following query:

```
SELECT DISTINCT EmployeeNumber
FROM ASSIGNMENT
WHERE HoursWorked > 50;
```

The result is:

EmployeeNumber	
1	8
2	10



## SQL for Data Retrieval: Subquery Example

Now, we can combine these two SQL statements by using a **subquery**, as follows:

```
SELECT  FirstName, LastName
FROM    EMPLOYER
WHERE   EmployeeNumber IN
        (SELECT  DISTINCT EmployeeNumber
         FROM    ASSIGNMENT
         WHERE   HoursWorked > 50);
```

The result of this expression is:

	FirstName	LastName
1	Tom	Jackson
2	Ken	Numoto



## SQL for Data Retrieval: Join Example

EmployeeNumber	FirstName	LastName	Department	Phone	Email
1	Mary	Jacobs	Administration	360-285-8110	Mjacobs@WPC.com
2	Rosalie	Jackson	Administration	360-285-8120	Rjackson@WPC.com
3	Richard	Bardalone	Legal	360-285-8210	Rbardalone@WPC.com
4	Tom	Caruthers	Accounting	360-285-8310	Tcaruthers@WPC.com
5	Heather	Jones	Finance	360-285-8420	Hjones@WPC.com
6	Mary	Abernathy	Finance	360-285-8410	MAbernathy@WPC.com
7	George	Smith	Human Resources	360-285-8510	Gsmith@WPC.com
8	Tom	Jackson	Production	360-287-8610	Tjackson@WPC.com
9	George	Jones	Production	360-287-8620	Gjones@WPC.com
10	Ken	Numoto	Marketing	360-287-8710	KNumoto@WPC.com
11	James	Nestor	Info Systems		JNestor@WPC.com
12	Rick	Brown	Info Systems	360-287-8820	RBrown@WPC.com

  

ProjectID	EmployeeNumber	HoursWorked
1000	1	30,00
1000	8	75,00
1000	10	55,00
1100	4	40,00
1100	6	45,00
1200	1	25,00
1200	2	20,00
1200	4	45,00
1200	5	40,00
1300	1	35,00
1300	8	80,00
1300	10	50,00
1400	4	15,00
1400	5	10,00
1400	6	27,50

  

ProjectID	Name	Department	MaxHours	StartDate	EndDate	Add New Field
1000	2008 Q3 Product Plan	Marketing	135,00	10-05-2008	15-06-2008	
1100	2008 Q3 Portfolio Analysis	Finance	120,00	05-07-2008	25-07-2008	
1200	2008 Q3 Tax Preparation	Accounting	145,00	10-08-2008	15-10-2008	
1300	2008 Q4 Product Plan	Marketing	150,00	10-08-2008	15-09-2008	
1400	2008 Q4 Portfolio Analysis	Finance	140,00	05-10-2008		



## SQL for Data Retrieval: JOIN...ON Example

```
SELECT  FirstName, LastName, HoursWorked
FROM    EMPLOYEE, ASSIGNMENT
WHERE   EMPLOYEE.EmployeeNumber =
        ASSIGNMENT.EmployeeNumber
ORDER BY EMPLOYEE.EmployeeNumber, ProjectID;
```

Using the JOIN . . . ON syntax, this query would be written as:

```
SELECT  FirstName, LastName, HoursWorked
FROM    EMPLOYEE JOIN ASSIGNMENT
        ON  EMPLOYEE.EmployeeNumber =
            ASSIGNMENT.EmployeeNumber
ORDER BY EMPLOYEE.EmployeeNumber, ProjectID;
```

	FirstName	LastName	HoursWorked
1	Mary	Jacobs	30.00
2	Mary	Jacobs	25.00
3	Mary	Jacobs	35.00
4	Rosalie	Jackson	20.00
5	Tom	Caruthers	40.00
6	Tom	Caruthers	45.00
7	Tom	Caruthers	15.00
8	Heather	Jones	40.00
9	Heather	Jones	10.00
10	Mary	Abernathy	45.00
11	Mary	Abernathy	27.50
12	Tom	Jackson	75.00
13	Tom	Jackson	80.00
14	Ken	Numoto	55.00
15	Ken	Numoto	50.00



## SQL for Data Retrieval: LEFT OUTER JOIN Example

- The OUTER JOIN syntax can be used to obtain data that exists in one table without matching data in the other table.



## SQL for Data Retrieval: LEFT OUTER JOIN Example

ProjectID	Name	Department	MaxHours	StartDate	EndDate	Add New Field
1000	2008 Q3 Product Plan	Marketing	135,00	10-05-2008	15-06-2008	
1100	2008 Q3 Portfolio Analysis	Finance	120,00	05-07-2008	25-07-2008	
1200	2008 Q3 Tax Preparation	Accounting	145,00	10-08-2008	15-10-2008	
1300	2008 Q4 Product Plan	Marketing	150,00	10-08-2008	15-09-2008	
1400	2008 Q4 Portfolio Analysis	Finance	140,00	05-10-2008		

ProjectID	EmployeeNumber	HoursWorked
1000	1	30,00
1000	8	75,00
1000	10	55,00
1100	4	40,00
1100	6	45,00
1200	1	25,00
1200	2	20,00
1200	4	45,00
1200	5	40,00
1300	1	35,00
1300	8	80,00
1300	10	50,00
1400	4	15,00
1400	5	10,00
1400	6	27,50

Course: DE4, Course Teacher: Dr. D. M. Akbar Hussain  
Department of Electronic Systems

3-71



## SQL for Data Retrieval: LEFT OUTER JOIN Example

```
SELECT Name, EmployeeNumber, HoursWorked
FROM PROJECT LEFT JOIN ASSIGNMENT
ON PROJECT.ProjectID = ASSIGNMENT.ProjectID;
```

	Name	EmployeeNumber	HoursWorked
1	2008 Q3 Product Plan	1	30.00
2	2008 Q3 Product Plan	8	75.00
3	2008 Q3 Product Plan	10	55.00
4	2008 Q3 Portfolio Analysis	4	40.00
5	2008 Q3 Portfolio Analysis	6	45.00
6	2008 Q3 Tax Preparation	1	25.00
7	2008 Q3 Tax Preparation	2	20.00
8	2008 Q3 Tax Preparation	4	45.00
9	2008 Q3 Tax Preparation	5	40.00
10	2008 Q4 Product Plan	1	35.00
11	2008 Q4 Product Plan	8	80.00
12	2008 Q4 Product Plan	10	50.00
13	2008 Q4 Portfolio Analysis	4	15.00
14	2008 Q4 Portfolio Analysis	5	10.00
15	2008 Q4 Portfolio Analysis	6	27.50

Course: DE4, Course Teacher: Dr. D. M. Akbar Hussain  
Department of Electronic Systems

3-72