Introduction to Sturctured QueryLanguage(SQL)

UCLA Extension
MGMNT X 414.61

Instructor

Instructor:

- Richard Patlan, M.A.
- Capital Programs, UCLA
- Consultant, Database Design and Website Development
- www.dbwebsite.com

Phone:

- Cell: 626-221-8435
- Work: 310-206-5908

E-Mail:

rpatlan@dbwebsite.com

- rpatlan@ucla.edu

Course Goals

Brief History of Structures Query Language

Understand and use Data Manipulation Language

Understand and use Data Definition Language

Introduction to SQL Language

Course Topics

- Overview of Structured Query Language (SQL):

  History of SQL and the American National Standards Institute (ANSI);

SQL Database Tables:

  Understanding Relational Database Management Systems;
  Creating tables, columns and rows;

Course Topics

SQL Data Manipulation Language (DML): Used to retrieve and update the contents of a database;

Data Definition Language (DDL): Used to create database objects such as tables, indexes, etc.

SQL Queries: Using SQL syntax to execute queries; and getting and using data result sets;

Readings

- SAMs Teach Yourself SQL in 10 Minutes
  By Ben Forta, Fourth Edition 2013

Free Software Download

- Microsoft SQL Server 2012 Express

SQL Server – Query Analyzer

- Load SQL Server 2012 – Load the SQL Server Management Studio application by clicking on the application icon or selecting it from the programs menu.

- Review Query Analyzer – This is the window pane in SQL Server where you write all your sql code

Overview

- Database Concepts
- Structured Query Language(SQL)

SQL – Database Review

- Databases History and Concepts

Database History

Database History

- 1960's – file processing systems: punch cards, paper tape, magnetic tape – sequential access and batch processing

- 1970s – Hierarchical and Network (legacy, some still used today)
- difficulties = hard to access data (navigational record-at-a-time procedures), limited data independence, no widely accepted theoretical model (unlike relational)

Database History

- 1974 -- Structured Query Language appeared.
- 1980s – Relational – E.F. Codd and others developed this theoretically well-founded model – all data represented in the form of tables – Oracle, DB2, Ingres

Database History

- 1986 -- IBM developed the first prototype of relational database and standardized by ANSI. The first relational database was released by Relational Software and its later becoming Oracle.

- 1987 -- Microsoft released SQL Server

Database History

- 1990s – Object-oriented, but some organizations have to handle large amounts of both structured and unstructured data, so Object-relational databases developed.

- 2000 and beyond – multi –tier, client-server, distributed environments, web-based, content-addressable storage, data mining

Database Concepts

Database Concepts

What is a database?

- A ***database (db)*** is an organized collection of data, typically stored in electronic format
  It allows you to input, manage, organize, and retrieve data quickly

  Traditional databases are organized by records (rows), fields (columns) stored in tables which are stored in the database files

Database Concepts

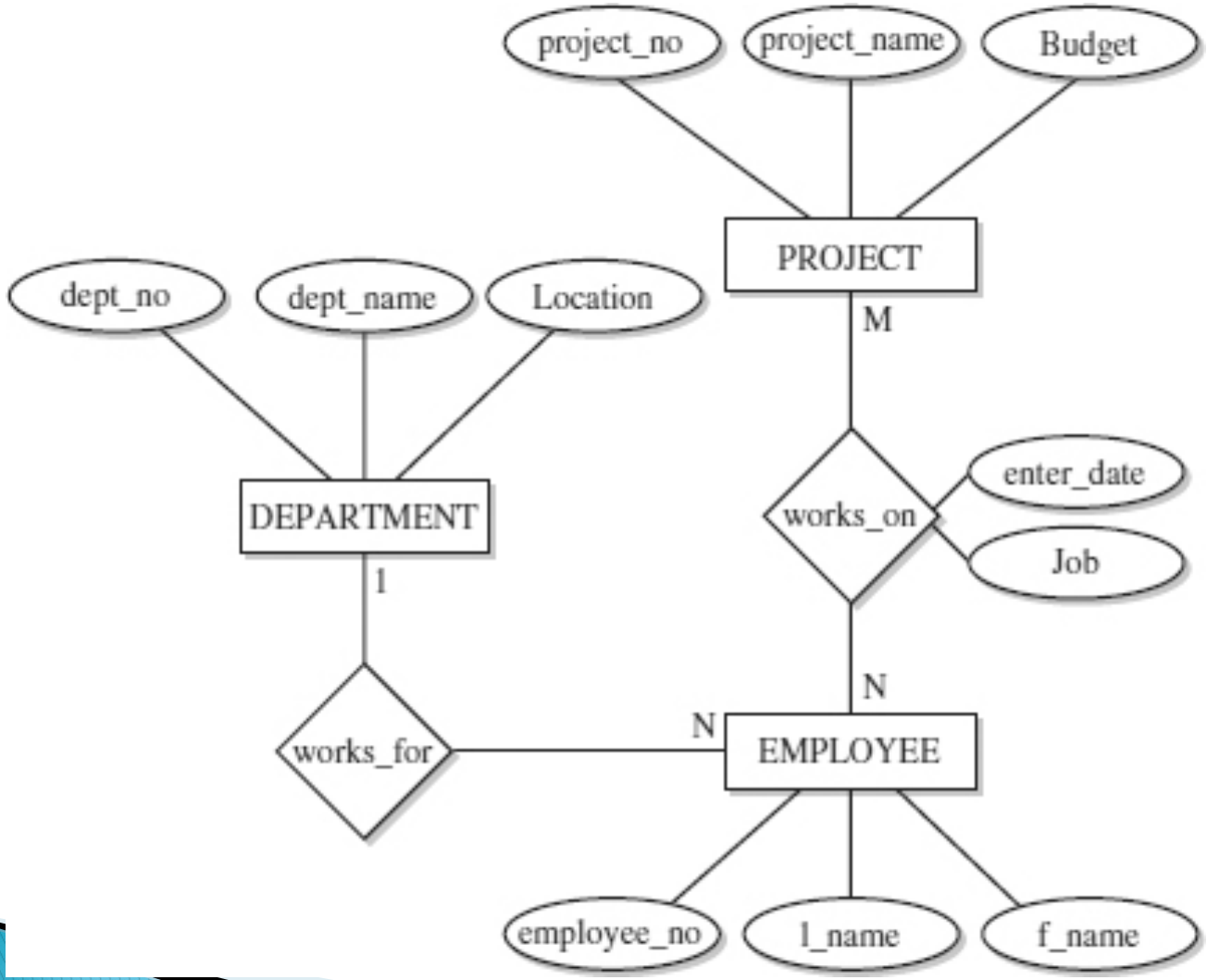
- A relational database system should model the real-world environment where it's used.
- The job of the database designer is to analyze the real-world system and then map it onto a relational database system.

ER Model

- To model a database and the relationships between its tables after a real-world system, you can use an *entity-relationship* (*ER*) *model* and a *relational model*.

- The ***entity-relationship (ER) model*** is a conceptual model. It is concerned with the logical nature of the data and what is being represented.

ER Model

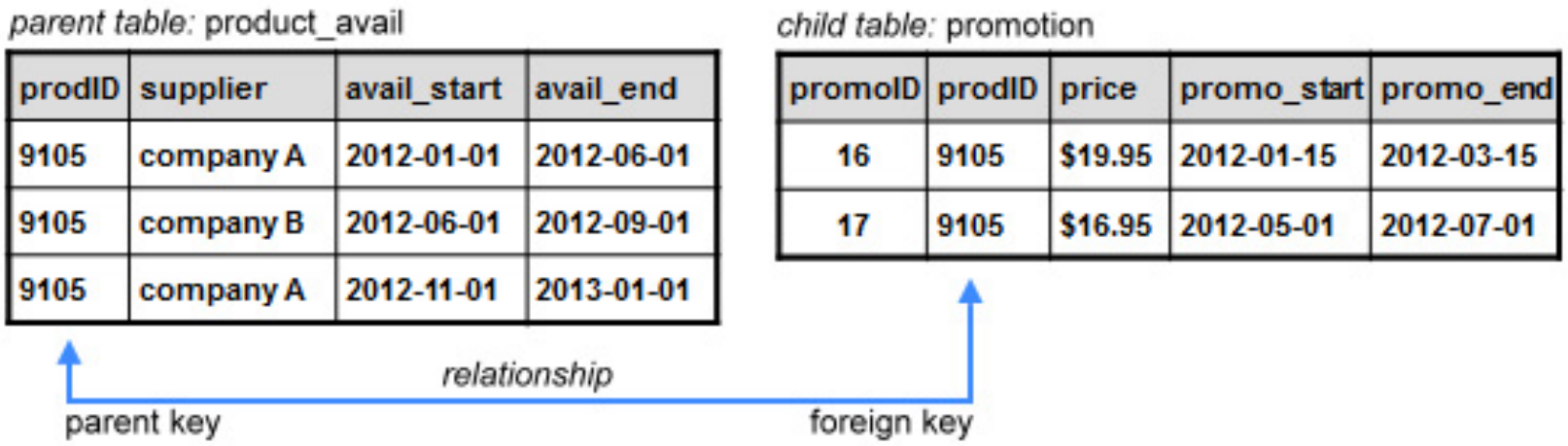

project_no
project_name
Budget
PROJECT
M
dept_no
dept_name
Location
DEPARTMENT
works_on
enter_date
Job
1
N
works_for
N
EMPLOYEE
employee_no
l_name
f_name

Database Concepts

- The *relational model* is an implementation model. It is concerned with the physical nature of the data and how the data will be represented in the database.

- Generally the conceptual models are concerned with the users view of the data and the implementation models are concerned with the developers view of the data

Relational Model

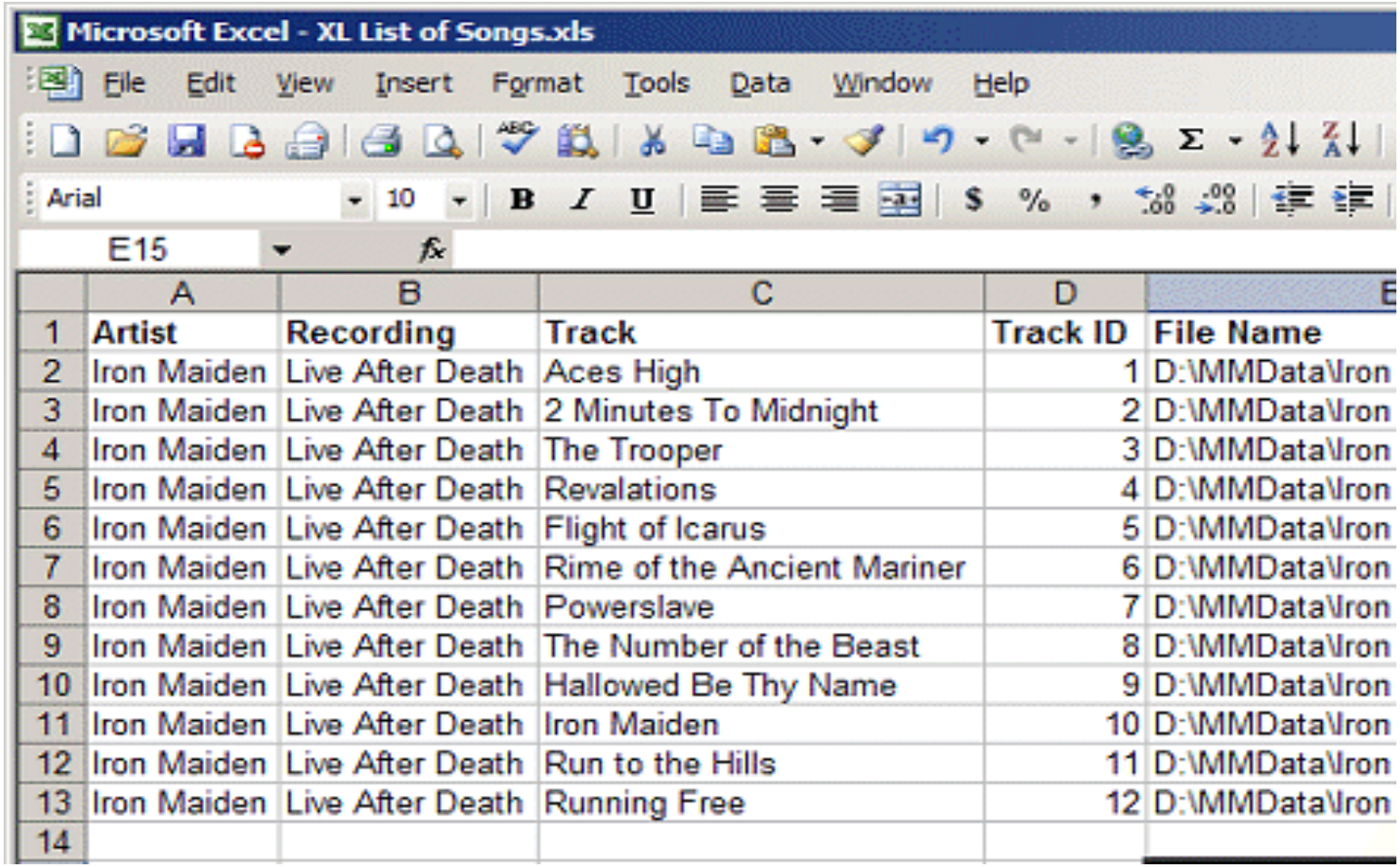


Database Concepts

Excel ‘tables’

- The idea of a table shouldn’t be new to you if you have used Excel, as that has rows and columns of information
- The structure of a SQL Server table is similar to that of an Excel spreadsheet

Excel ‘tables’



	A	B	C	D	E
1	**Artist**	**Recording**	**Track**	**Track ID**	**File Name**
2	Iron Maiden	Live After Death	Aces High	1	D:\MMData\Iron
3	Iron Maiden	Live After Death	2 Minutes To Midnight	2	D:\MMData\Iron
4	Iron Maiden	Live After Death	The Trooper	3	D:\MMData\Iron
5	Iron Maiden	Live After Death	Revalations	4	D:\MMData\Iron
6	Iron Maiden	Live After Death	Flight of Icarus	5	D:\MMData\Iron
7	Iron Maiden	Live After Death	Rime of the Ancient Mariner	6	D:\MMData\Iron
8	Iron Maiden	Live After Death	Powerslave	7	D:\MMData\Iron
9	Iron Maiden	Live After Death	The Number of the Beast	8	D:\MMData\Iron
10	Iron Maiden	Live After Death	Hallowed Be Thy Name	9	D:\MMData\Iron
11	Iron Maiden	Live After Death	Iron Maiden	10	D:\MMData\Iron
12	Iron Maiden	Live After Death	Run to the Hills	11	D:\MMData\Iron
13	Iron Maiden	Live After Death	Running Free	12	D:\MMData\Iron
14					

Tables

- A database ***table*** is a collection of rows and columns that is used to organize information about a single topic.
- Each row within a table corresponds to a single record and contains several attributes that describe the row.
- These tables are stored in databases. Next slide shows a sample table

Tables

EmployeeID	LastName	FirstName	Department
100	Smith	Bob	IT
101	Jones	Susan	Marketing
102	Adams	John	Finance

Database Concepts

- Tables must have unique names
- A table has properties that defines how data is stored in them, how it is broken up, how individual pieces of information are named, etc.
- These properties that define a table is called the Schema

Database Concepts

- Tables are comprised of one or more fields or columns
- A column is a single field in a table
- Each column stores a specific type of data

Database Concepts

- Each column definition indicates whether or not it can contain *null values*.
- A null value indicates that the value of the column is unknown.

Database Concepts

- A column can also be defined with a *default value*. Then, that value is used if another value isn't provided when a row is added to the table.

- A column can also be defined as an *identity column*. An identity column is a numeric column whose value is generated automatically when a row is added to the table.

Database Concepts

- Data is stored in rows. Records and rows mean the same thing.
- A row contains a set of values for a single instance of the entity, such as one invoice or one vendor.

Structured Query Language

What is SQL?

- The acronym, SQL, stands for Structured Query Language.
- It is commonly pronounced as 'Sequel'
- SQL is simply pronounced, 'S Q L'. You say each letter individually. It is an acronym, not an abbreviation.

What is SQL?

- SQL is not a proprietary language used by specific database vendors. Almost every major DBMS supports SQL
- Learning this language will allow you to interact with just about any database.
- SQL (Structured Query Language) is a database computer language designed for managing data in relational database management systems (RDBMS).

SQL – Structured Query language

A Database Computer Language designed for Managing Data in Relational Database Management Systems (RDBMS)

Query Examples:

```
• insert into STUDENT (Name , Number, SchoolId)
  values ('John Smith', '100005', 1)

• select SchoolId, Name from SCHOOL

• select * from SCHOOL where SchoolId > 100

• update STUDENT set Name='John Wayne' where StudentId=2

• delete from STUDENT where SchoolId=3
```

We have 4 different Query Types: **INSERT**, **SELECT**, **UPDATE** and **DELETE**

What is SQL?

- SQL can execute queries against a database
- SQL can retrieve data from a database
- SQL can insert records in a database
- SQL can update records in a database
- SQL can delete records from a database

What is SQL?

- SQL can create new databases
- SQL can create new tables in a database
- SQL can create stored procedures in a database
- SQL can create views in a database
- SQL can set permissions on tables, procedures, and views

SQL is ANSI Standard

- There are many different versions of the SQL language.
- Most of the SQL database programs also have their own proprietary extensions in addition to the SQL standard!
- But to be in compliance with the ANSI standard, they must support the same major keywords in a similar manner.

SQL is ANSI Standard

- SQL is an American National Standards Institute (ANSI) standard computer language

- SQL is an ANSI standard computer language for accessing and manipulating database systems.

SQL is ANSI Standard

There are lots of different database systems, or DBMS – Database Management Systems, such as:

MS Access,
DB2, Informix,
MS SQL Server,
Oracle,
Sybase,
etc.

DML

Data Manipulation Language (DML)

- The **Data Manipulation Language (DML)** is the subset of SQL used to add, update and delete data.

- The acronym **CRUD** refers to all of the major functions that need to be implemented in a relational database application to consider it complete.

Data Manipulation Language (DML)

- Each letter in the acronym can be mapped to a standard SQL statement:

Operation	SQL	Description
Create	INSERT INTO	inserts new data into a database
Read (Retrieve)	SELECT	extracts data from a database
Update	UPDATE	updates data in a database
Delete (Destroy)	DELETE	deletes data from a database

Data Manipulation Language (DML)

This include:

- SELECT
- UPDATE
- DELETE
- INSERT
- WHERE
- and others

SQL Basics

With SQL you can:

- Retrieve data from a database querying using the SQL SELECT Statement
- Manipulate data and generate information

SQL Syntax and Conventions

- SQL statements are represented as text.
- SQL statements are independent of text lines.
- SQL statements can be placed on one text line or on multiple text lines.

SQL Syntax and Conventions

- SQL statements are case sensitive or case in-sensitive
- SQL can be upper or lower case

  - If SQL is case sensitive then case for statements matters
  - If SQL is case in-sensitive then case for statements does not matter.

SQL Queries

- With SQL, we can query a database and have a result set returned.
- A query like this:
  - *SELECT LastName FROM Persons*

SQL Queries

- A Case Sensative query :
  - *SELECT LastName FROM Persons*
- A Case In-Sensative query :
  - *SELECT lastName FROM persons*

SQL Queries

Select LastName
FROM Persons

Gives a result set like this:

LastName
Hansen
Svendson
Pettersen

- Note: Some database systems require a semicolon at the end of the SQL statement.

SQL – SELECT

The SELECT statement is probably the most used SQL command.

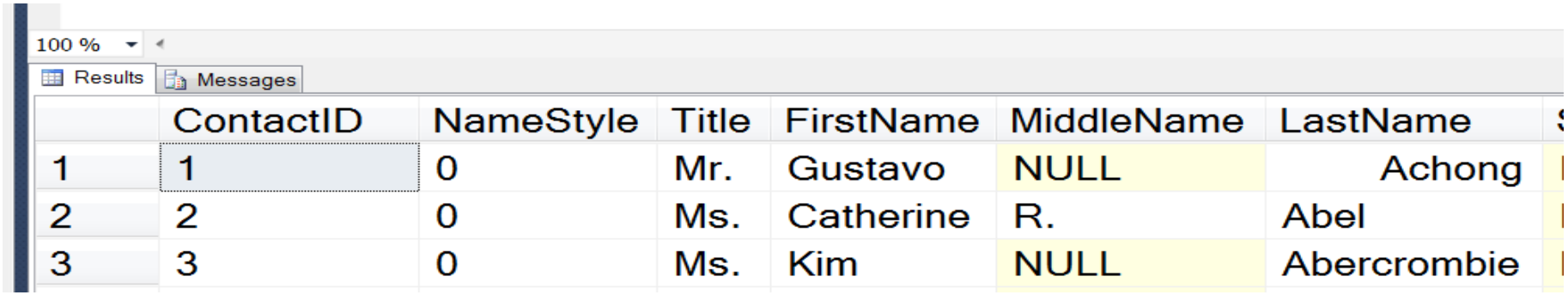
The SELECT statement is used for retrieving rows from the database and enables the selection of one or many rows or columns from one or many tables in the database.

We will use the Contact table as an example.

SQL – SELECT

Example:

```
SELECT *
FROM
Contact
```



100 %

Results | Messages

	ContactID	NameStyle	Title	FirstName	MiddleName	LastName
1	1	0	Mr.	Gustavo	NULL	Achong
2	2	0	Ms.	Catherine	R.	Abel
3	3	0	Ms.	Kim	NULL	Abercrombie

SQL – SELECT

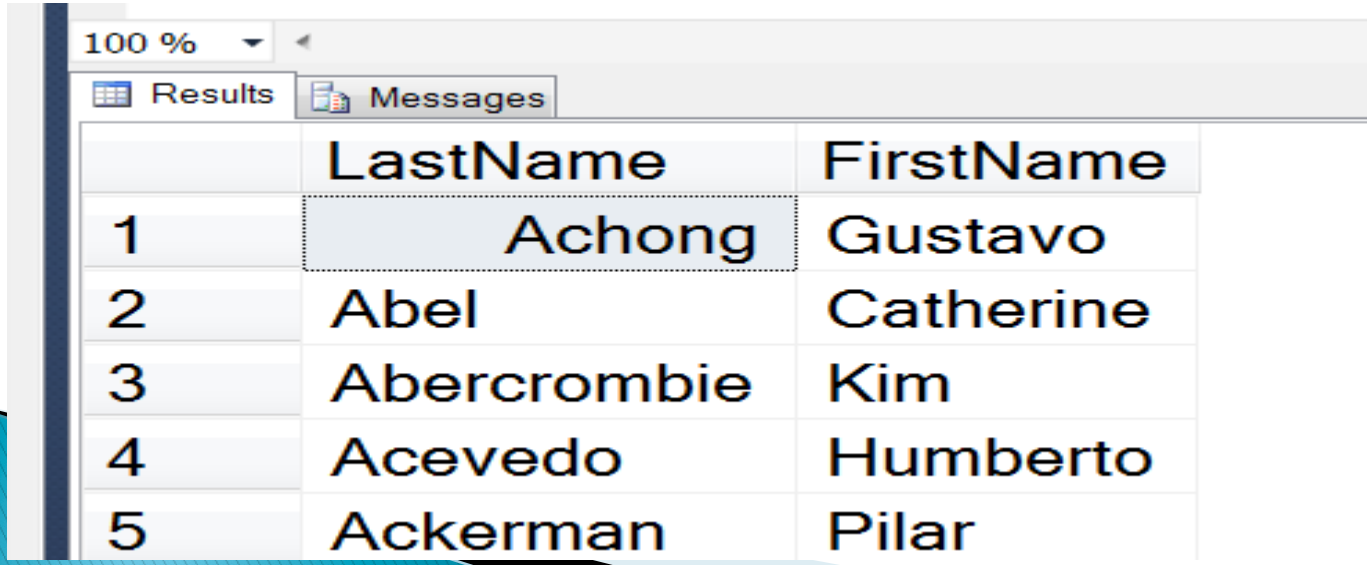
This simple example gets all the data in the table Contact.

The symbol “*” is used when you want to get all the columns in the table.

SQL – SELECT

If you only want a few columns, you may specify the names of the columns you want to retrieve, example:

- SELECT LastName, FirstName
- FROM
- Contact



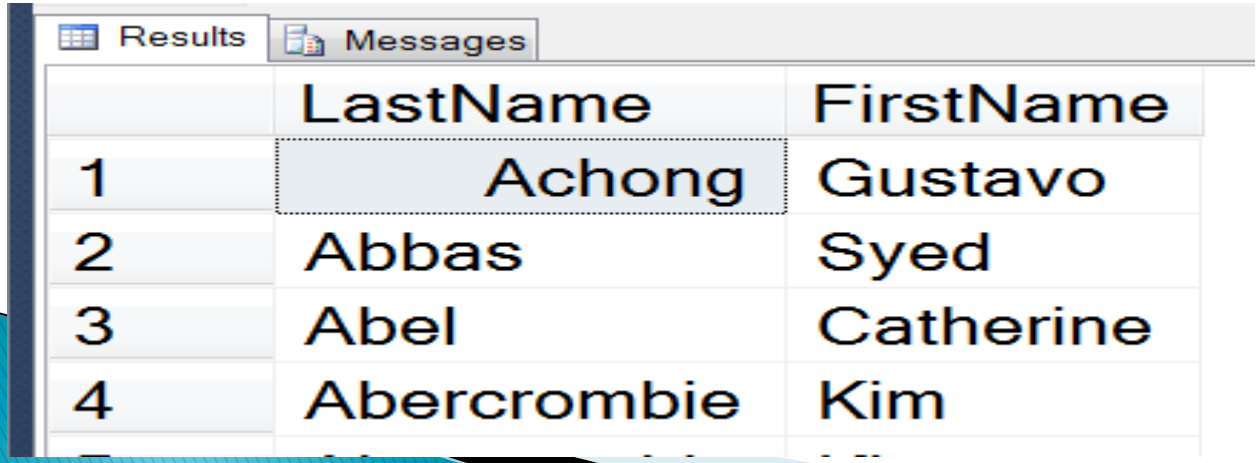
	LastName	FirstName
1	Achong	Gustavo
2	Abel	Catherine
3	Abercrombie	Kim
4	Acevedo	Humberto
5	Ackerman	Pilar

SQL – ORDER BY Keyword

If you want the data to appear in a specific order you need to use the “order by” keyword.

Example:

```
SELECT LastName,FirstName
FROM
Contact
ORDER BY LastName
```

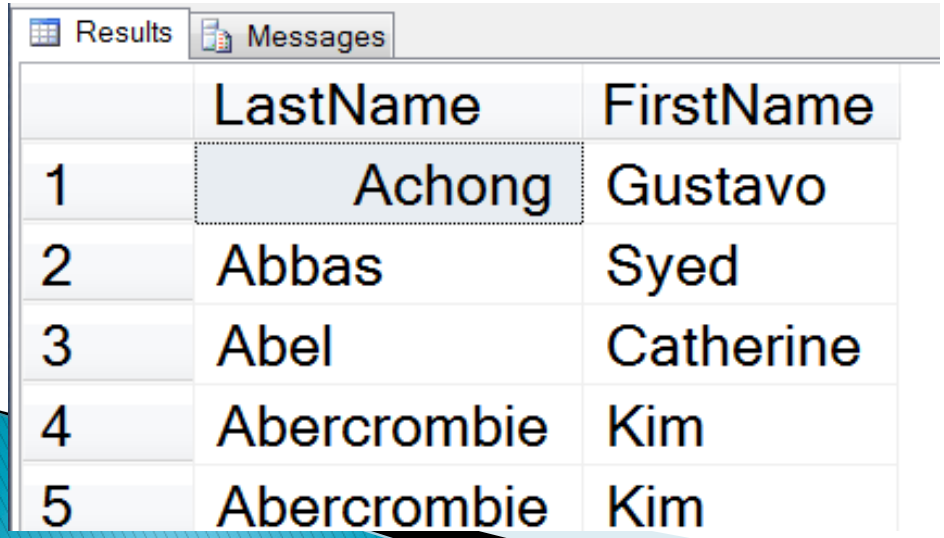


	LastName	FirstName
1	Achong	Gustavo
2	Abbas	Syed
3	Abel	Catherine
4	Abercrombie	Kim

SQL – ORDER BY Keyword

You may also sort by several columns, e.g. like this:

```
SELECT LastName, FirstName
FROM
Contact
ORDER BY LastName, FirstName
```



Results | Messages

	LastName	FirstName
1	Achong	Gustavo
2	Abbas	Syed
3	Abel	Catherine
4	Abercrombie	Kim
5	Abercrombie	Kim

SQL – ORDER BY Keyword

If you use the “order by” keyword, the default order is ascending (“asc”).

If you want the order to be opposite, i.e., descending, then you need to use the “desc” keyword.

SQL – ORDER BY Keyword

SELECT LastName,FirstName
FROM
Contact

ORDER BY LastName desc

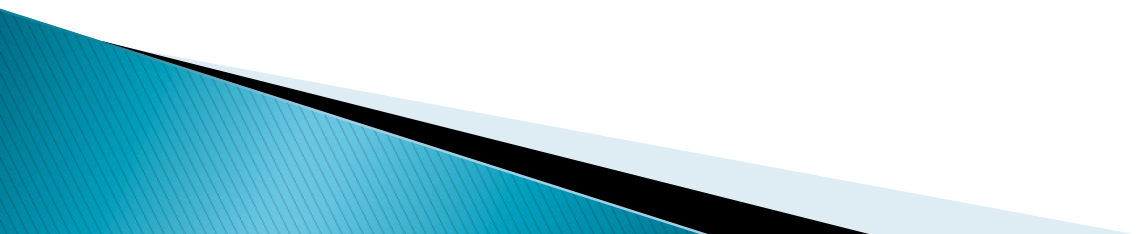
	LastName	FirstName
1	Zwilling	Michael
2	Zwilling	Michael
3	Zukowski	Jake
4	Zugelder	Judy
5	Zubaty	Carla

SQL - SELECT DISTINCT

In a table, some of the columns may contain duplicate values.

This is not a problem, however, sometimes you will want to list only the different (distinct) values in a table.

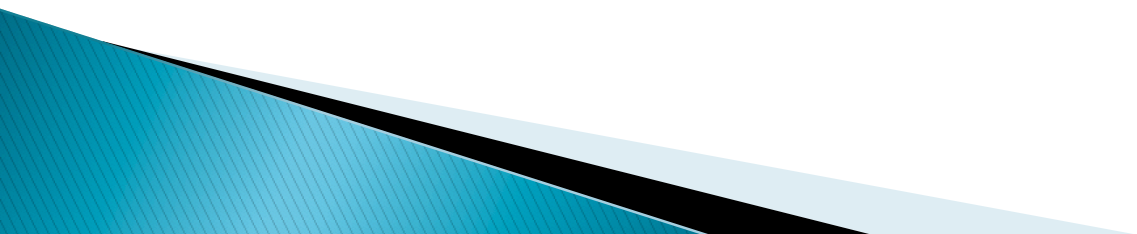
The DISTINCT keyword can be used to return only distinct (different) values.



SQL – SELECT DISTINCT

The syntax is as follows:

Select Distinct <column_names> From <table_names>



SQL – SELECT DISTINCT

Example:

SELECT Distinct LastName
FROM
Contact

Results | Messages

	LastName
1	Barlow
2	Billstrom
3	Hassall
4	Groncki
5	Verdad

SQL – WHERE Clause

The WHERE clause is used to extract only those records that fulfill a specified criterion.

The syntax is as follows:

Select <column_names>
From <table_name>
Where <column_name> operator value

SQL – WHERE Clause

The WHERE clause can use different types of OPERATORS:

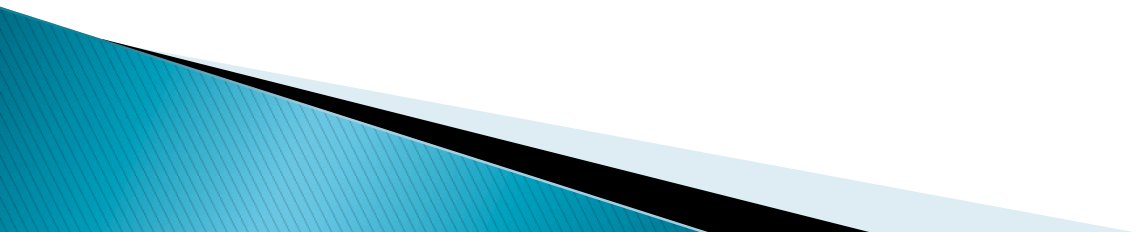
- Comparison

SQL – Comparison OPERATORS

Operator	Example	Defined	Result
=, IS	5 = 5	5 equal to 5?	True
!=, IS NOT	7 != 2	7 IS NOT (!=) equal to 2?	True
<	7 < 4	7 less than 4?	False
>	7 > 4	greater than 4?	True
<=	7 <= 11	Is 7 less than or equal to 11?	True
>=	7 >= 11	Is 7 greater than or equal to 11?	False

SQL – Comparison OPERATORS

Equality involves comparing two values. To do so requires the use of the (<), (>), or (=) special characters. Does X = Y? Is Y < X? These are both questions that can be answered using a SQL Equality Operator expression.

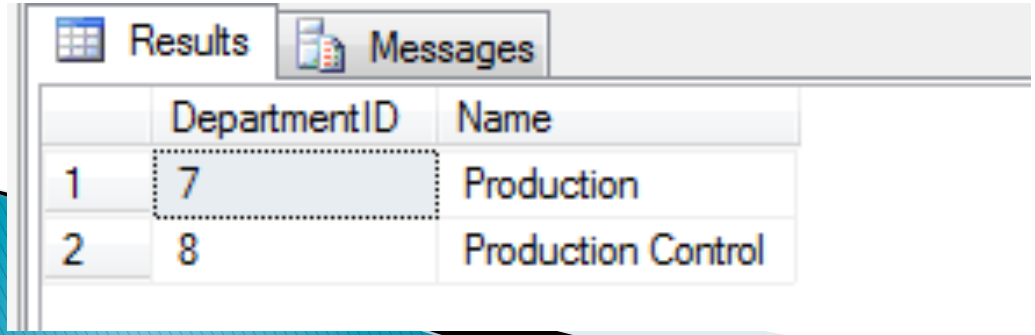


SQL – Equals operator

The following example uses the Equals operator to return all rows in the `Department` table in which the value in the `GroupName` column is equal to the word 'Manufacturing'.

Example:

```
SELECT DepartmentID, Name
FROM Department
WHERE GroupName = 'Manufacturing';
```

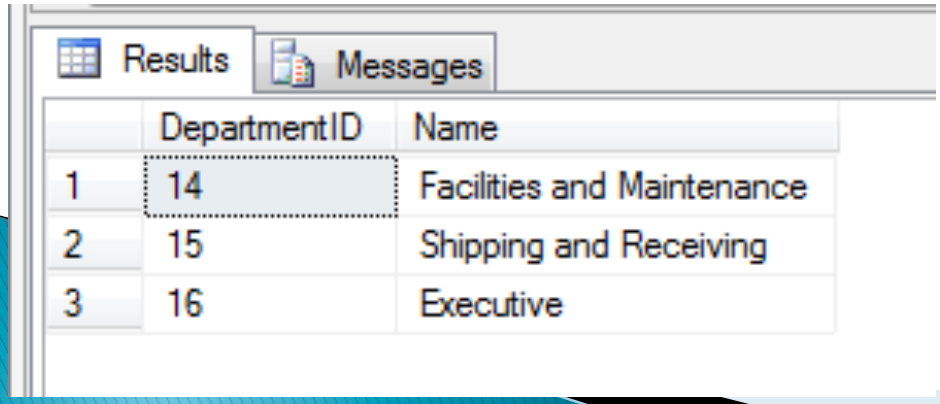


SQL – Greater Than operator

The following example returns all rows in the `Department` table that have a value in `DepartmentID` that is greater than the value 13.

Example:

```
SELECT DepartmentID, Name
FROM Department
WHERE DepartmentID > 13
ORDER BY DepartmentID;
```

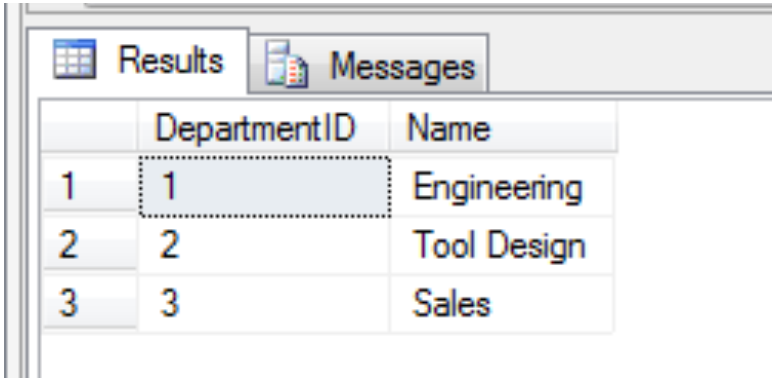


SQL – Less Than or Equal operator

- The following example returns all rows in the `Department` table that have a value in `DepartmentID` that is less than or equal to the value 3.

Example:

```
SELECT DepartmentID, Name
FROM Department
WHERE DepartmentID <= 3
ORDER BY DepartmentID;
```



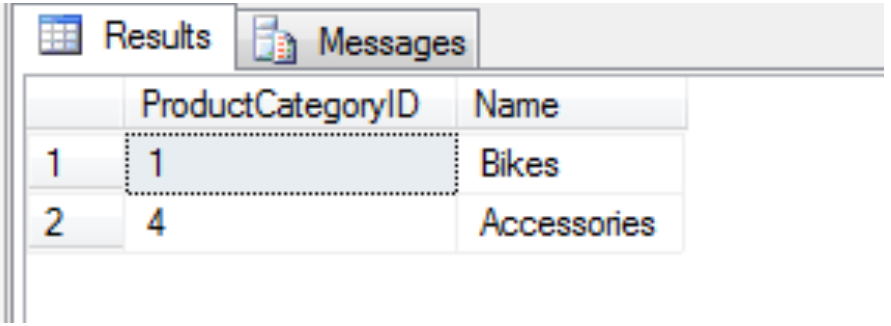
	DepartmentID	Name
1	1	Engineering
2	2	Tool Design
3	3	Sales

SQL – Not Equal operator

The following example returns all rows in the `ProductCategory` table that do not have value in `ProductCategoryID` that is equal to the value 3 or the value 2.

Example:

```
SELECT ProductCategoryID, Name
FROM ProductCategory
WHERE ProductCategoryID <> 3
AND ProductCategoryID <> 2;
```

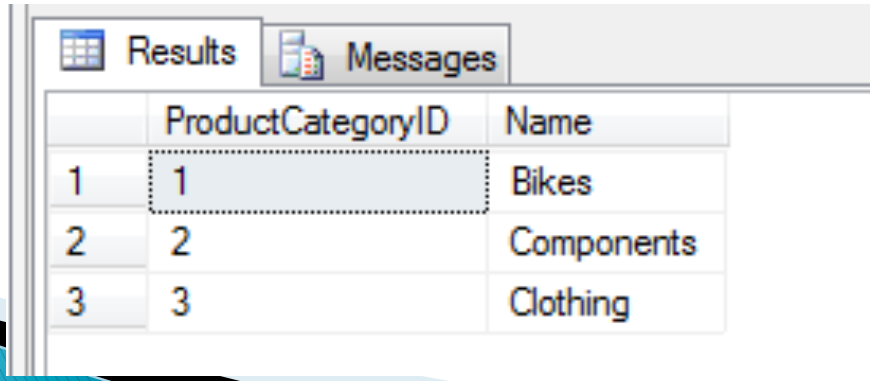


SQL – Between operator

The following example returns all rows in the `ProductCategory` table that have value in `ProductCategoryID` between 1 and 3.

Example:

```
SELECT ProductCategoryID, Name
FROM ProductCategory
WHERE ProductCategoryID Between 1 AND 3
```



	ProductCategoryID	Name
1	1	Bikes
2	2	Components
3	3	Clothing

SQL – Is Null operator

The following example returns all rows in the Contact table Where Middle Name value is NULL.

Example:

```
SELECT *
FROM Contact
WHERE MiddleName IS NULL
```

Results | Messages

	ContactID	NameStyle	Title	FirstName	MiddleName	LastName
1	1	0	Mr.	Gustavo	NULL	Achong
2	3	0	Ms.	Kim	NULL	Abercrombie
3	4	0	Sr.	Humberto	NULL	Acevedo
4	5	0	Sra.	Pilar	NULL	Ackerman
5	9	0	Mr.	Jay	NULL	Adams
6	14	0	Mr.	François	NULL	Ferrier