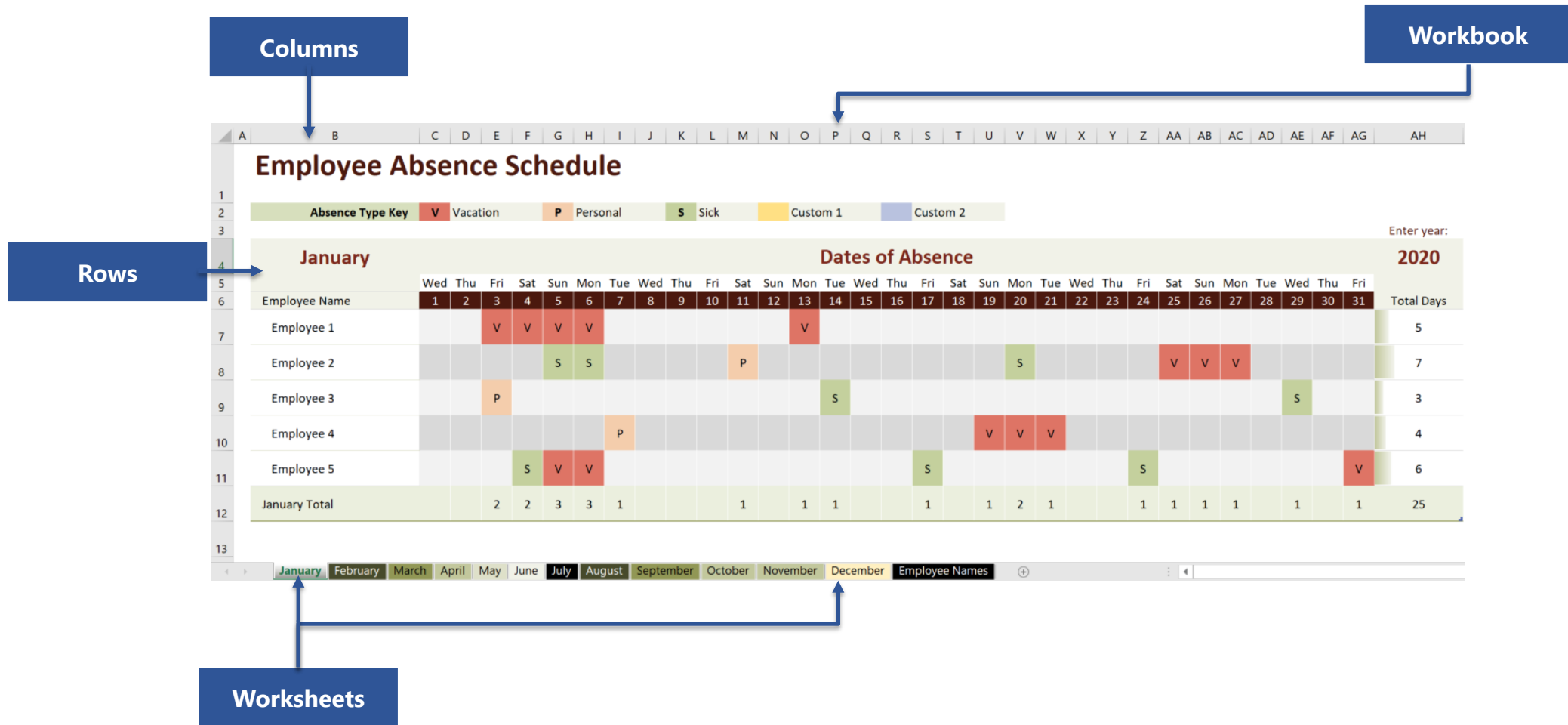


PART 2: SQL Basics



PART 2: SQL Basics

Database

Columns

Workbook

Rows

Columns

Worksheets

Employee Absence Schedule

Absence Type Key: V Vacation, P Personal, S Sick, Custom 1, Custom 2

January		Dates of Absence																															Enter year: 2020
Employee Name	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	Total Days	
Employee 1			V	V	V	V							V																		5		
Employee 2					S	S					P									S					V	V	V				7		
Employee 3			P										S															S			3		
Employee 4							P												V	V	V										4		
Employee 5				S	V	V										S							S						V		6		
January Total			2	2	3	3	1				1		1	1		1		1	1	2	1		1	1	1	1		1		1	25		

Tables

Worksheets

PART 2: SQL Basics

A Database can have one or more tables.

Order Table

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10248	90	5	1996-07-04	3
10249	81	6	1996-07-05	1
10250	84	4	1996-07-08	2
10251	84	3	1996-07-08	1
10252	76	4	1996-07-09	2

Shipper Table

ShipperID	ShipperName	Phone
1	Speedy Express	(503) 555-9831
2	United Package	(503) 555-3199
3	Federal Shipping	(503) 555-9931

Customer Table

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

PART 2: SQL Basics

Structured Query Language (SQL)

SQL is a programming language used for interacting with relational database management system (RDBMS).

- Create, retrieve, update, delete data
- Create, manage database
- Design, create database tables
- Perform administrative tasks, such as user management, import/export data, security etc.

SQL implementation may vary between different RDBMS slightly.

In this part, we will use [w3school.com/sql/default.asp](https://www.w3school.com/sql/default.asp) to demonstrate the basic SQL queries.

PART 2: SQL Basics

SELECT Query 101

Retrieve data from a relational database

Query

- A query in itself is just a statement which declares what data we are looking for.
- SQL statements are **NOT** case sensitive. **Select** is the same as **SELECT**.
- Table name and column name, however are case sensitive, eg. **CustomerName** <> **CUSTOMERNAME**
- A **semicolon** in a statement is a standard way to separate SQL statements that allow more than one SQL statement.

Syntax

```
SELECT column1, column2, ...  
FROM table_name;
```

Example

```
SELECT CustomerName,City FROM Customers;
```

```
SELECT * FROM Customers;
```

Result

CustomerName	City
Alfreds Futterkiste	Berlin
Ana Trujillo Emparedados y helados	México D.F.
Antonio Moreno Taquería	México D.F.
Around the Horn	London
Berglunds snabbköp	Luleå

PART 2: SQL Basics

SELECT Query 101

Retrieve data from a relational database

WHERE Clause

- A where clause is used to filter records.
- The clause is applied to each row by checking specific column values to determine whether it should be included.
- More complex clause can be constructed by joining various **AND** or **OR**.

Syntax

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition;
```

Example

```
SELECT * FROM Customers  
WHERE Country='Mexico';
```

Result

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
13	Centro comercial Moctezuma	Francisco Chang	Sierras de Granada 9993	México D.F.	05022	Mexico
58	Pericles Comidas clásicas	Guillermo Fernández	Calle Dr. Jorge Cash 321	México D.F.	05033	Mexico
80	Tortuga Restaurante	Miguel Angel Paolino	Avda. Azteca 123	México D.F.	05033	Mexico

PART 2: SQL Basics

Query with Constraints

Extract only those records that fulfill a specified condition

Logical Operator

Operator	Description
AND	The AND operator allows the existence of multiple conditions in an SQL statement's WHERE clause.
BETWEEN	The BETWEEN operator is used to search for values that are within a set of values, given the minimum value and the maximum value.
IN	The IN operator is used to compare a value to a list of literal values that have been specified.
LIKE	The LIKE operator is used to compare a value to similar values using wildcard operators.
NOT	The NOT operator reverses the meaning of the logical operator with which it is used. Eg. NOT EXISTS, NOT BETWEEN, NOT IN etc. This is negate operator.
OR	The OR operator is used to combine multiple conditions in an SQL statement's WHERE clause.
IS NULL	The NULL operator is used to retrieve NULL values

Syntax

```
SELECT column, another_column, ...  
FROM tables  
WHERE condition  
    AND/OR another_condition  
    AND/OR ...
```

Example

```
SELECT * FROM Customers  
WHERE Country='Germany' AND (City='Berlin' OR City='München');
```

Result

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
25	Frankenversand	Peter Franken	Berliner Platz 43	München	80805	Germany

PART 2: SQL Basics

SQL Operators - LIKE

LIKE operator is used in a WHERE clause to search for a specified pattern in a column.

LIKE Operator

There are two wildcards often used in conjunction with the LIKE operator:

- % The percent sign represents zero, one, or multiple characters
- _ The underscore represents a single character

Operator	Condition	Example
LIKE	exact string comparison	col_name LIKE "ABC"
NOT LIKE	exact string inequality comparison	col_name NOT LIKE "ABCD"
%	match a sequence of zero or more characters (only with LIKE or NOT LIKE)	col_name LIKE "%AT%" (matches "AT", "AT&T", "CAT")
_	match a single character (only with LIKE or NOT LIKE)	col_name LIKE "AN_" (matches "AND", but not "AN")

Syntax

```
SELECT * FROM Customers
WHERE CustomerName LIKE 'a%';
```

Result

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK

PART 2: SQL Basics

SQL Operators - IN

IN operator allows you to specify multiple values in a WHERE clause.

IN Operator

- IN operator is a shorthand for multiple OR conditions. Below two queries are identical.

```
SELECT CustomerName , Country from Customers
where Country IN ('Mexico', 'UK')
```

```
SELECT CustomerName , Country from Customers
where Country = 'Mexico' or Country = 'UK'
```

- NOT IN is to test whether an expression is not in a list.

Operator	Condition	Example
IN (...)	String exists in a list	col_name IN ("A", "B", "C")
NOT IN (...)	String does not exist in a list	col_name NOT IN ("D", "E", "F")

Syntax

```
SELECT column_name(s)
FROM table_name
WHERE column_name IN (value1, value2, ...);
```

Example

```
SELECT CustomerID, CustomerName, Country FROM Customers
WHERE Country IN ('Germany', 'France', 'UK');
```

Result

CustomerID	CustomerName	Country
1	Alfreds Futterkiste	Germany
4	Around the Horn	UK
6	Blauer See Delikatessen	Germany
7	Blondel père et fils	France
9	Bon app'	France
11	B's Beverages	UK
16	Consolidated Holdings	UK
17	Drachenblut Delikatessend	Germany

PART 2: SQL Basics

SQL Operators - BETWEEN

BETWEEN operator selects values within a given range.

BETWEEN Operator

- The values can be **numbers**, **text**, or **dates**.
- BETWEEN operator is inclusive: begin and end values are included.
- BETWEEN can be used with the NOT operator to create a NOT BETWEEN condition.

Syntax

```
SELECT column_name(s)
FROM table_name
WHERE column_name BETWEEN value1 AND value2;
```

Example

```
SELECT ProductName, Unit, Price FROM Products
WHERE Price BETWEEN 10 AND 20;
```

Result

ProductName	Unit	Price
Chais	10 boxes x 20 bags	18
Chang	24 - 12 oz bottles	19
Aniseed Syrup	12 - 550 ml bottles	10
Genen Shouyu	24 - 250 ml bottles	15.5
Pavlova	32 - 500 g boxes	17.45
Sir Rodney's Scones	24 pkgs. x 4 pieces	10
NuNuCa Nuß-Nougat-Creme	20 - 450 g glasses	14

PART 2: SQL Basics

Sorting Query

ORDER BY keyword sorts the records in ascending order by default.

ORDER BY Clause

- ORDER BY sorts each row alpha-numerically based on the specified column's value.
- ORDER BY keyword sorts the records in ascending order by default.

Syntax

```
SELECT column1, column2, ...  
FROM table_name  
ORDER BY column1, column2, ... ASC|DESC;
```

Example

```
SELECT * FROM Customers  
ORDER BY Country, CustomerName;
```

Result

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
12	Cactus Comidas para llevar	Patricio Simpson	Cerrito 333	Buenos Aires	1010	Argentina
54	Océano Atlántico Ltda.	Yvonne Moncada	Ing. Gustavo Moncada 8585 Piso 20-A	Buenos Aires	1010	Argentina
64	Rancho grande	Sergio Gutiérrez	Av. del Libertador 900	Buenos Aires	1010	Argentina
20	Ernst Handel	Roland Mendel	Kirchgasse 6	Graz	8010	Austria
59	Piccolo und mehr	Georg Pippis	Geislweg 14	Salzburg	5020	Austria
50	Maison Dewey	Catherine Dewey	Rue Joseph-Bens 532	Bruxelles	B-1180	Belgium
76	Suprêmes délices	Pascale Cartrain	Boulevard Tirou, 255	Charleroi	B-6000	Belgium

PART 2: SQL Basics

Filtering Query

LIMIT will reduce the number of rows to return, and the optional OFFSET will specify where to begin counting the number rows from.

LIMIT & OFFSET Clause

- **LIMIT 10** would return the first 10 rows matching the SELECT criteria.
- Statement **LIMIT 3 OFFSET 1** means it skips the first record that would normally be returned and instead return the second, third, and fourth records.
- ORDER BY clause are commonly used with the LIMIT and OFFSET clauses, which are a useful optimization to indicate to the database the subset of the results you care about.

Syntax

```
SELECT column_name(s)
FROM table_name
WHERE condition
LIMIT number;
```

Example

```
SELECT * FROM Customers LIMIT 3;
```

Result

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico

PART 2: SQL Basics

Unique Query

DISTINCT statement is used to return only distinct (different) values.

DISTINCT Clause

- A column often contains many duplicate values; and sometimes you only want to list the different (distinct) values.
- DISTINCT clause consider NULL to be a unique value in SQL.
- DISTINCT can be used with aggregates: COUNT, AVG, MAX, etc.

Syntax

```
SELECT DISTINCT column1, column2, ...  
FROM table_name;
```

Example

```
SELECT DISTINCT Country FROM Customers;
```

Result

Number of Records: 21

Country
Germany
Mexico
UK
Sweden
France
Spain
Canada