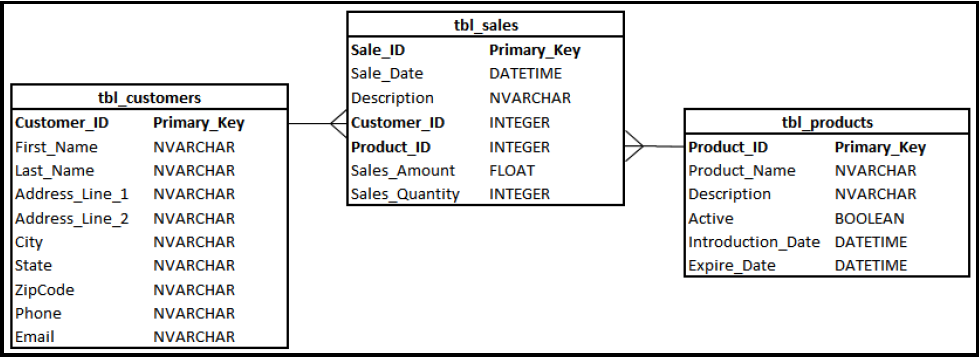
**SQL Lab 1**

**What is a database?** A technological solution to record and retrieve data in an effective and efficient way

We use the customer\_sales.db database from textbook1 to get to know some simple SQL syntaxes.

This is the UML of this database:



It has three tables: customers, sales, and products.

**Four main functions in SQL?** Select, Insert, update, delete

## Select:

The SELECT statement is used to select data from a database.

The data returned is stored in a result table, called the result-set.

|  |
| --- |
|  |

Examples:

|  |
| --- |
| --Q1  select First\_Name, Last\_Name, City  from tbl\_customers; |

|  |
| --- |
| --Q2  select \*  from tbl\_customers; |

## Commenting out in SQL:

* **Inline comment**: --
* **Multiline comment**: /\* whatever is here is commented. \*/

## Distinct:

It is used to return only distinct (different) values.

Inside a table, a column often contains many duplicate values; and sometimes you only want to list the different (distinct) values.

|  |
| --- |
|  |

Examples: Compare

|  |
| --- |
| --Q3  Select Distinct State  from tbl\_customers; |

|  |
| --- |
| --Q4  Select State  from tbl\_customers; |

## Where – AND/OR/NOT:

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

|  |
| --- |
|  |

Examples:

|  |
| --- |
| --Q5  Select First\_Name, Last\_Name, State  from tbl\_customers  Where State = 'NJ'; |

|  |
| --- |
| --Q6  Select First\_Name, Last\_Name, State  from tbl\_customers  Where State is 'NJ'; |

The WHERE clause can be combined with AND, OR, and NOT operators.

The AND and OR operators are used to filter records based on more than one condition

* The AND operator displays a record if all the conditions separated by AND are TRUE.
* The OR operator displays a record if any of the conditions separated by OR is TRUE.
* The NOT operator displays a record if the condition(s) is NOT TRUE.

Examples:

|  |
| --- |
| --Q7  Select First\_Name, Last\_Name, State  from tbl\_customers  Where State is 'NJ' and Last\_Name='Winer'; |

## Order by

|  |
| --- |
|  |

|  |
| --- |
| --Q8  select \*  from tbl\_customers  Order by City; |

|  |
| --- |
| --Q9  select \*  from tbl\_customers  Order by City Desc, Customer\_ID Asc; |

* Asc stands for Ascending
* Desc stands for Descending

**Example 1:**

List the Last Name and address of Customers who are from TX, NJ

|  |
| --- |
| --E1  select Last\_Name, Address\_Line\_1, State  from tbl\_customers  Where State = 'TX' OR State = 'NJ'; |

OR

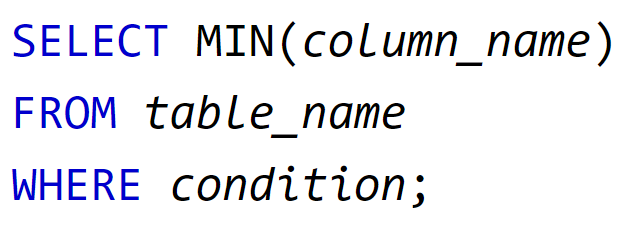
|  |
| --- |
| --E1  select Last\_Name, Address\_Line\_1, State  from tbl\_customers  Where State in ('TX','NJ'); |

**Example 2:**

Find the one customer name and list their last name, address, and the city which has purchased a product on 4/18/2016.

|  |
| --- |
| --E2  select Last\_Name, Address\_Line\_1, State  from tbl\_customers  where Customer\_ID in (select Customer\_ID  from tbl\_sales  where Sale\_Date = '4/18/2016'); |

MIN(), MAX(), AVE(), SUM(), COUNT()



|  |
| --- |
| --Q10  select MAX(Sales\_amount)  from tbl\_sales |

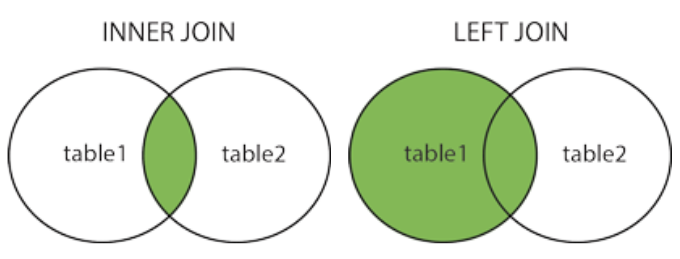
|  |
| --- |
| --Q11  select Count(Customer\_ID)  from tbl\_customers  Where State in ('TX','NJ'); |

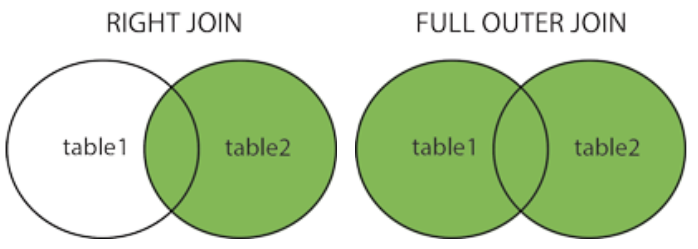
Join:

Used to combine rows from two or more tables, based on a related column between them.

Different types of the JOINs in SQL:

* **(INNER) JOIN:** Returns records that have matching values in both tables
* **LEFT (OUTER) JOIN:** Returns all records from the left table, and the matched records from the right table
* **RIGHT (OUTER) JOIN:** Returns all records from the right table, and the matched records from the left table
* **FULL (OUTER) JOIN:** Returns all records when there is a match in either left or right table





Inner join is the one that will be mostly used:

|  |
| --- |
| SELECT column\_name(s) FROM table1 INNER JOIN table2 ON table1.column\_name=table2.column\_name; |

|  |
| --- |
| --Q12  select \*  from tbl\_customers inner join tbl\_sales  on tbl\_customers.Customer\_ID = tbl\_sales.Customer\_ID |

**Example 3:**

List the last names and addresses of customers that their sales have ever purchased through the internet.

|  |
| --- |
| --E3  select distinct First\_Name, Last\_Name, Address\_Line\_1  from tbl\_customers inner join tbl\_sales  on tbl\_customers.Customer\_ID = tbl\_sales.Customer\_ID  Where Description == 'Internet Purchase' |

**Example 4:**

Write a query that lists all of the customers who has ever purchased ‘Shoe Shine’

|  |
| --- |
| --E4  select distinct First\_Name, Last\_Name, Address\_Line\_1  from tbl\_customers  inner join tbl\_sales  on tbl\_customers.Customer\_ID = tbl\_sales.Customer\_ID  inner join tbl\_products  on tbl\_products.Product\_ID = tbl\_sales.Product\_ID  Where Product\_Name = 'Shoe Shine' |

**Example 7:**

Find a list of customers who has purchased anything sweet.

|  |
| --- |
| Select Distinct CustomerName  from Customers join Orders  on Customers.CustomerID = Orders.CustomerID  join OrderDetails  on Orders.OrderID = OrderDetails.OrderID  join Products  on OrderDetails.ProductID = Products.ProductID  join Categories  on Products.CategoryID = Categories.CategoryID  Where Description like '%sweet%' |

Like

|  |
| --- |
|  |

Like Operator is used in a WHERE clause to search for a specified pattern in a column.

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

Find all the customers whose country’s name starts with B

|  |
| --- |
| SELECT \* FROM Customers  Where Country like 'B%' |

Find all the customers whose city’s name ends with n

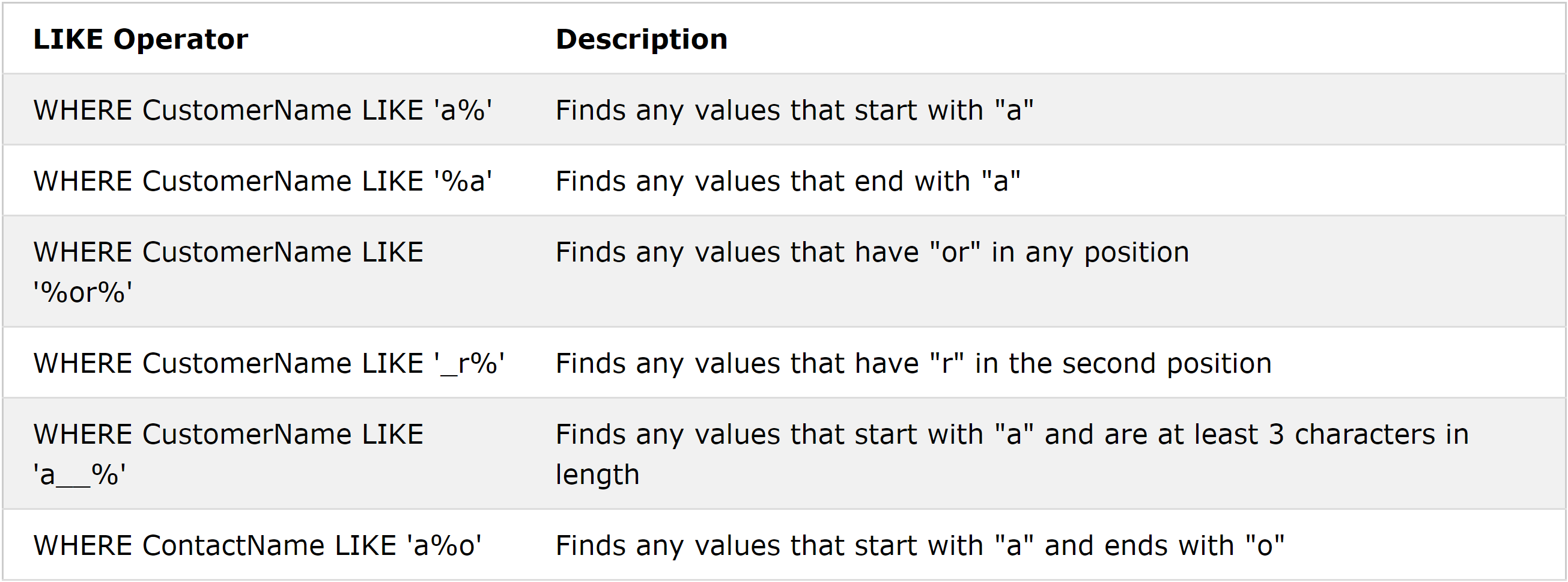
|  |
| --- |
| SELECT \* FROM Customers  Where City like '%n' |

Find all the employees whose note include the word psychology

|  |
| --- |
| SELECT \* FROM Employees  where Notes like '%psychology%' |

Find all the employees whose birthdate is in month 9

|  |
| --- |
| SELECT \* FROM Employees  where BirthDate like '\_\_\_\_-09%' |



**Example 3:**

Write a query that count the number of custoemrs who live in the 97XXX USA zip code.

|  |
| --- |
| SELECT count(\*) FROM Customers  Where Country is 'USA' AND PostalCode like '97%' |

Group By:

Group by groups rows that have the same values into summary rows, like "find the number of customers in each country".

The GROUP BY statement is often used with aggregate functions (COUNT, MAX, MIN, SUM, AVG) to group the result-set by one or more columns.

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
| Select Country, count(CustomerID) as CountryCount  From Customers  Group by Country  Order by CountryCount Desc |

**Example 8:**

List customers name and their total numbers of orders

|  |
| --- |
| SELECT CustomerName, Count(OrderID) as Count  FROM Customers  INNER JOIN Orders  On Customers.CustomerID = Orders.CustomerID  GROUP BY Customers.CustomerID  Order by Count desc |

**Example 9:**

List customers names and their total number times that have ordered something sweet

|  |
| --- |
| Select Customers.CustomerID, CustomerName, Count(Orders.OrderID) as SweetCount  from Customers join Orders  on Customers.CustomerID = Orders.CustomerID  join OrderDetails  on Orders.OrderID = OrderDetails.OrderID  join Products  on OrderDetails.ProductID = Products.ProductID  join Categories  on Products.CategoryID = Categories.CategoryID  Where Description like '%sweet%'  group by (Customers.CustomerID)  Order by SweetCount Desc |

Having Clause:

The HAVING clause was added to SQL because the WHERE keyword could not be used with aggregate functions.

|  |
| --- |
|  |

|  |
| --- |
| Select Country, count(CustomerID) as CountryCount  From Customers  Group by Country  Having CountryCount>5  Order by CountryCount Desc |

**Example 10:**

List all the countries and all the total amount of sales coming from them.

|  |
| --- |
| Select Country, sum(Unit\*Price) as Sales  From Customers join Orders  on Customers.CustomerID = Orders.CustomerID  join OrderDetails  on OrderDetails.OrderID = Orders.OrderID  join Products  on OrderDetails.ProductID = Products.ProductID  group by Country  Order by Sales Desc |

**Example 11:**

List all customers from countries that has more than 10,000 total sales.

|  |
| --- |
| Select CustomerName from customers  Where Country in (  Select Country  From Customers join Orders  on Customers.CustomerID = Orders.CustomerID  join OrderDetails  on OrderDetails.OrderID = Orders.OrderID  join Products  on OrderDetails.ProductID = Products.ProductID  group by Country  Having sum(Unit\*Price) > 999  ) |

Assignment:

**Challenge 1:**

List shippers’ name, phone number and the number of orders that they have handled

|  |
| --- |
| SELECT ShipperName, Phone, Count(OrderID) AS Count  From Shippers  INNER JOIN Orders  ON Shippers.ShipperID = Orders.ShipperID  GROUP BY (Shippers.ShipperID)  ORDER BY Count Desc |

**Challenge 2:**

**List customer names with the total number of Items that they have bought**

|  |
| --- |
| SELECT CustomerName, Sum(Quantity)  FROM Customers  INNER Join Orders  ON Customers.CustomerID = Orders.CustomerID  INNER JOIN OrderDetails  ON OrderDetails.OrderID = Orders.OrderID  INNER JOIN Products  ON OrderDetails.ProductID = Products.ProductID  Group By Customers.CustomerID  Order by Sum(Quantity) Desc |

**Challenge 3:**

Draw up a list that shows the number of times each employee has used different shippers

|  |
| --- |
| SELECT Shippers.ShipperID, Orders.ShipperID, LastName, ShipperName, count(OrderID) AS TotalNumberOfOrders  From Employees  INNER JOIN Orders  ON Orders.EmployeeID = Employees.EmployeeID  INNER JOIN Shippers  ON Shippers.ShipperID = Orders.ShipperID  GROUP BY Orders.EmployeeID, Orders.ShipperID  ORDER BY LastName, TotalNumberOfOrders desc |

**Challenge 4:**

List the country names that has had a customer whose total purchase has been than 5,000

|  |
| --- |
| SELECT distinct Country  FROM Customers  INNER Join Orders  ON Customers.CustomerID = Orders.CustomerID  INNER JOIN OrderDetails  ON OrderDetails.OrderID = Orders.OrderID  INNER JOIN Products  ON OrderDetails.ProductID = Products.ProductID  Group By Customers.CustomerID  Having sum(Quantity\*price) > 5000  Order by Sum(Quantity) Desc |