**Basic SQL statements**  
**Exercises**  
  
Select  
Write a statement to get all the columns from the Customers table.  
Select \* From Customers;  
  
Write a statement that will select the City column from the Customers table.Select City from Customers;  
  
Select all the different values from the Country column in the Customers table.  
Select \* Distinct Country from Customers;  
  
Select all records where the City column has the value "Berlin".  
Select \* from Customers  
Where City= ’ Berlin ’;  
  
Use the NOT keyword to select all records where City is NOT "Berlin".  
Select \* From Customers  
Where Not City = ’ Berlin ’ ;  
  
Select all records where the City column has the value 'Berlin' *and* the PostalCode column has the value 12209  
Select \* From Customers  
Where City = ‘Berlin’  
And PostalCode = 12209;  
  
Select all records where the City column has the value 'Berlin' or 'London'.  
Select \* From Customers  
Where City = ‘ Berlin ’  
OR City = ‘ London ’  
  
Select all records from the Customers table, sort the result alphabetically by the column City.  
Select \* From Customers  
Order BY City DESK ;  
  
Select all records from the Customers table, sort the result alphabetically, first by the column Country, then, by the column City.  
Select \* From Customers  
Order By Country, City ;  
  
Insert a new record in the Customers table.  
Insert INTO Customers (CustomerName, City , Address)  
VALUES ( ‘Dima Stankevich’, ‘Warszawa’, ‘Bytnara 7’ )  
Select all records from the Customers where the PostalCode column is empty.  
Select \* From Customers  
Where PostalCode is Not Null;  
  
Update the City column of all records in the Customers table.  
UPDATE Customers  
SET City = ’OSLO”;  
  
Set the value of the City columns to 'Oslo', but only the ones where the Country column has the value "Norway".  
UPDATE Customers  
SET City= ‘Oslo’  
WHERE Country = ‘Norway’  
  
Update the City value *and* the Country value.  
UPDATE CUSTOMERS  
SET City = ‘Oslo’ ,  
Country = ‘Norway’  
Where CustomerID = 32;  
  
Delete all the records from the Customers table where the Country value is 'Norway'.  
DELETE FROM Customers  
Where City= ‘Norway’ ;  
  
Use the MIN function to select the record with the smallest value of the Price column.  
Select MIN(Price)  
From Products  
  
Use an SQL function to select the record with the highest value of the Price column.  
Select Max(Price)  
FROM Products  
  
Use the correct function to return the number of records that have the Price value set to 18.  
Select Count (\*)  
From Products  
Where Price = 18  
  
Use an SQL function to calculate the average price of all products.  
Select AVG(Price)  
From Products;  
  
  
Use an SQL function to calculate the sum of all the Price column values in the Productstable.  
Select Sum(Price)  
From Products;  
  
Select all records where the value of the City column starts with the letter "a".  
Select \* From Customers  
WHERE City LIKE ‘a%’;  
  
Select all records where the value of the City column *ends* with the letter "a".  
Select \* From Customers  
Where City LIKE ‘%a’;  
  
Select all records where the value of the City column contains the letter "a".  
Select \* From Customers  
Where City like ‘%a%’  
  
Select all records where the value of the City column starts with letter "a" and ends with the letter "b".  
Select \* From Customers  
Where City LIKE ‘a%b’  
  
Select all records where the value of the City column does NOT start with the letter "a".

Select \* From Customers  
Where City Not LIKE “a%’;  
  
Select all records where the *second* letter of the City is an "a".  
Select \* From Customers

Where City LIKE ‘\_a%’;  
  
Select all records where the first letter of the City is an "a" or a "c" or an "s".  
Select \* From Customers  
Where City LIKE [acs];  
  
Select all records where the first letter of the City starts with anything from an "a" to an "f".  
Select \* from Customers  
Where City LIKE [a-f];  
  
Select all records where the first letter of the City is NOT an "a" or a "c" or an "f".

Select \* From Customers  
Where City LIKE ‘[!acf];  
  
Use the IN operator to select all the records where Country is NOT "Norway" and NOT "France".  
Select \* From Customers  
Where City Not LIKE in (‘Norway’, ‘France’)

Use the BETWEEN operator to select all the records where the value of the Price column is NOT between 10 and 20.  
Select \* From Products  
Where Price NOT BETWEEN 10 and 20;  
  
Use the BETWEEN operator to select all the records where the value of the ProductName column is alphabetically between 'Geitost' and 'Pavlova'.  
Select \* From Products  
Where Productname BETWEEN ‘Geitost’ and ‘Pavlova’;  
  
When displaying the Customers table, make an ALIAS of the PostalCode column, the column should be called Pno instead.  
Select CustomerName,  
Address,  
PostalCode AS Pno  
From Customers;

Insert the missing parts in the JOIN clause to join the two tables Orders and Customers, using the CustomerID field in both tables as the relationship between the two tables.  
Select \* From Orders  
Left JOIN Customers  
On Orders.CustomerID=Customers.CustomerID;  
  
Choose the correct JOIN clause to select all records from the two tables where there is a match in both tables.  
Select \* From Orders  
INNER Join Customers  
ON Orders.CustomerID.Customers.CustomerID;  
  
Choose the correct JOIN clause to select all the records from the Customers table plus all the matches in the Orders table.  
Select From Orders;  
Right Join Customers  
Orders.CustomerID=Customers.CustomerID;