

SQL STATEMENT SUMMARY

Format of Insert Statement:

```
INSERT INTO <table> (<field>)  
VALUES <value1>, <value2>, ... , <valuen>  
//Insert a record with the values to the fields listed
```

Format of an Insert with a Select

```
INSERT INTO tblVisits (PrisonerID, VisitorName, VisitLength, VisitDate)  
SELECT PrisonerID, 'Joan', VisitLength, NOW() FROM tblVisits  
WHERE PrisonerID = 3 AND VisitorName = 'Susan'  
//Use a SELECT statement in place of VALUES. Make sure the fields match the fields in the first line.
```

Format of Delete Statement: (use with caution)

```
DELETE FROM <table>  
WHERE <condition> AND/OR <condition>  
//Delete the record with the field that fits the condition
```

Updating Data in a Table

```
UPDATE <table>  
SET <field> = <new value>  
WHERE <field> = <existing value>  
//Change the value in the fields of the record that fits the condition to the new value
```

Format of Order Statement (Sorting)

```
SELECT <field(s)>  
FROM <table>  
ORDER BY <field1>, ... , <fieldn>  
//Order the records alphabetically by the first field then the next field
```

```
SELECT <field(s)>  
FROM <table>  
ORDER BY <field1> DESC, ... , <fieldn> DESC  
//Order the records alphabetically in descending order by the first field then the next field
```

Selecting Data in a Table

```
SELECT <field(s)>  
FROM <table>  
//Limit the amount of fields on the table
```

```
SELECT *  
FROM <table>  
//Show all fields of the table
```

Format to See Specific Records

```
SELECT <field(s)>  
FROM <table>  
WHERE <condition> // Show the records that fit the condition
```

```
SELECT <field(s)>  
FROM <table>  
WHERE <condition> LIKE <%pattern>  
//The records where the pattern exists at the end of the value  
LIKE <pattern%>  
//The records where the pattern exists at the beginning of the value  
LIKE <%pattern%>  
//The records where the pattern exists anywhere in the value
```

FROM <table>
WHERE <condition> AND <condition>
//The records where both the conditions are true

SELECT <field(s)>
FROM <table>
WHERE <condition> OR <condition>
//The records where either of the conditions are true

SELECT <field(s)>
FROM <table>
WHERE <field> BETWEEN <value1> AND <value2>
//The records where the field has values between the two values specified

SELECT <field(s)>
FROM <table>
WHERE <condition>
LIMIT <value>
//Only the records before the limit value are shown

SELECT <fields>
FROM <table>
WHERE <condition>
GROUP BY <field>
HAVING (<condition1>, <condition2>...)
//Having is for the conditions for the new result set
NB* use only when using GROUP BY

SELECT DISTINCT <fields>
FROM <table>
//Removes all duplicates

Embedded Queries

SELECT Name
FROM tblTeachers
WHERE Salary = (SELECT MAX(Salary) FROM tblTeachers)
//Criteria for the WHERE statement is another SQL statement

SELECT Name
FROM tblTeachers
WHERE Salary > (SELECT AVG(Salary) FROM tblTeachers)

Format for Word Manipulation

SELECT LEFT(<field>, <value>) AS <field>
FROM <table>
//characters from left until the value

SELECT RIGHT(<field>, <value>) AS <field>
FROM <table>
//all characters from right until the value from right

SELECT MID(<field>, <value1>, <value2>) AS <field>
FROM <table>
//all characters from value 1 until the amount of value 2 e.g. (3,2) will give you the 3rd and 4th values

SELECT LENGTH(<field>) AS <field>
FROM <table>
//give the length of the field

```
SELECT CONCAT(<value1>,<value2>,<value3>) AS <field>
FROM <table>
```

//gives the values combined together as one value

Format for Date Manipulation

```
SELECT MONTH(<field>) AS <field>
FROM <table>
```

//gives the month of the date

```
SELECT DAY(<field>) AS <field>
FROM <table>
```

//gives the day of the date

```
SELECT YEAR(<field>) AS <field>
FROM <table> //gives the year of the date
```

```
SELECT NOW() AS <field>
FROM <table>
```

//gives the current date from the computer clock

Format for Number Manipulation

```
SELECT AVG(<field>) AS <field>
FROM <table>
```

//gives the average of the records for that field

```
SELECT COUNT(*)
FROM <table>
```

WHERE <condition>

//Counts the amount of records in the table with the condition

```
SELECT MAX(<field>) AS <field>
FROM <table>
```

//Show the record with the biggest value in the field

```
SELECT MIN(<fields>) AS <field>
FROM <table>
```

//Show the record with the smallest value in the field

```
SELECT SUM(<field>) AS <field>
FROM <table>
```

//Show total sum of the records in the field

```
SELECT <field>, COUNT(*) AS <field>
FROM <table>
```

WHERE <condition>

GROUP BY <field>

//Show the amount of records in every field in the group by condition

```
SELECT ROUND(<field>,<value>) AS <field>
FROM <table>
```

//Rounds the field off to the "valueth" decimal place e.g. (3.456,1) => 3.5

```
SELECT RAND( ) AS <field>
FROM <table>
```

//Gives a random number from 0 to 0.9999 ...

```
SELECT MOD(<value>,<value>) AS <field>
FROM <table>
```

//Gives the remainder of the first value after dividing by the second value, same as the % for java


```
SELECT INT(<value>,<value>) AS <field>
FROM <table>
```

//Gives the integer value of a number after truncating the decimals.

```
SELECT TRUNCATE(<field manipulated>)
FROM <table>
```

//Gives the value in the same format as the field

```
SELECT <field>
FROM <table>
WHERE <field> IN (<value>, <value> ...)
```

//Same as saying <field> = <value> OR <field> = <value> ...

Format for Table Joins

```
SELECT <table>.<field>
FROM <table1>, <table2>
WHERE <table1>.<field> = <table2>.<field>
```

//Inner join by just using where conditions

```
SELECT <table>.<field>
FROM <table1>
INNER JOIN <table2> ON <table1>.<field> = <table2>.<field>
```

//Inner join with full format and the order of the tables do not matter

```
SELECT <table>.<field>
FROM <table1>
LEFT OUTER JOIN <table2> ON <table1>.<field> = <table2>.<field>
```

//Outer join, table 1 contains info and the second has null value

```
SELECT <table>.<field>
FROM <table1>
RIGHT OUTER JOIN <table2> ON <table1>.<field> = <table2>.<field>
```

//Outer join, table 2 contains info and the first has null value

```
SELECT <table>.<field>
FROM <table1>
FULL JOIN <table2> ON <table1>.<field> = <table2>.<field>
```

//Outer join, either table has info and either can have null value

Order of the Select Statement

```
SELECT DISTINCT <field>
FROM <table>
WHERE <condition>
GROUP BY <field>
HAVING <condition>
ORDER BY <field>
LIMIT <value>
```