

## SQL Summary – Grade 10. Version One.

**SELECT** <field(s)>

**FROM** <table(s)>

**WHERE** condition(s)

**GROUP BY** expression

**HAVING** condition

**ORDER BY** expression

**Arithmetic function:** INT( ), ROUND( ), TRUNCATE( ). Formats the single parameter within the brackets. RND - a single random number.

**Random numbers:** RND(Upper bound – Lower bound ) + Lower bound. Gives a random number from lower bound to upper bound eg 1 to 50.

**Date and time:** DATE, YEAR, MONTH, DAY, TIME, HOUR, MINUTE. Used with a field of the data type “date”. NOW( ) yields today from the PC.

**Aggregate functions:** MAX( ), MIN( ), AVG( ), SUM( ), COUNT( ). Returns a single result based on a group of rows.

**Compound conditions:** NOT, AND, OR

**More conditions:** BETWEEN .. AND, IN and NOT IN, LIKE, NULL

**Quotes.** Regular quotes for string data. Hash symbols for date/time . #2019/05/23 9:33:00#. Boolean (True) – no quotes.

**String manipulation:** LEFT(x), RIGHT(x), MID(x, y), LEN, and “&” which is the concatenation operator in Ms Access (joins strings)

**NOTE:** Ms Access is case insensitive (unlike Java). We use upper case and lower case to make our queries more readable.

## Using a table of Coffee types as an example . . . .

ProductID	ProductName	CountryOfOrigin	Roast	SizeInGrams	CostPrice	Click
1	Gold	Ethiopia	Dark	250	R38.00	
2	Gold	Ethiopia	Dark	1000	R140.00	
3	Blue	Limu	Medium	250	R31.00	
4	Blue	Limu	Medium	500	R58.00	
5	Blue	Limu	Medium	1000	R114.00	
6	Blue	Limu	Medium	2000	R200.00	
7	Green	Kenya	Dark	500	R60.00	
8	Green	Kenya	Dark	1000	R112.00	
9	Red	Malawi	Light	250	R40.00	
10	Purple	Rwanda	Medium	250	R36.00	
11	Purple	Rwanda	Medium	1000	R108.00	
12	Purple	Rwanda	Medium	2000	R196.00	
13	Yellow	Kenya	Dark	250	R42.00	
14	Yellow	Kenya	Dark	500	R79.00	
15	Yellow	Kenya	Dark	1000	R150.00	
16	Yellow	Kenya	Dark	2000	R240.00	
*	(New)			0	R0.00	

### SELECT ... FROM ...

1 SELECT \* FROM tblCoffee . . . . “ \* “ wildcard selects everything – all fields.

2 SELECT \* FROM tblCoffee ORDER BY ProductName ASC . . . . **ASC (increasing) is the default, DESC (decreasing)**

3 SELECT \* FROM tblCoffee WHERE ProductName = (G\*) . . . Only products that start with the letter “G”.

### THE BEST ... THE 5 BEST ... THE WORST ... THE HIGHEST ... THE MOST (EXPENSIVE) (CHEAPEST)

4 SELECT **TOP 1** CostPrice FROM tblCoffee ORDER BY CostPrice; . . . gives us the lowest price.

5 SELECT **TOP 1** CostPrice FROM tblCoffee ORDER BY CostPrice DESC; . . . gives us the highest price.

6 SELECT **TOP 5** CostPrice FROM tblCoffee ORDER BY CostPrice; . . . gives us the 5 lowest prices.

**SELECT ONLY CERTAIN FIELDS FROM THE TABLE.**

7 SELECT ProductName, CountryOfOrigin, Roast FROM tblCoffee

**SELECT ONLY CERTAIN FIELDS and SORTING BY ONE FIELD.**

8 SELECT ProductName, CountryOfOrigin, Roast FROM tblCoffee ORDER BY CountryOfOrigin

**SELECT VALUES THAT ARE UNIQUE EG COUNTRY OF ORIGIN – WE WANT EACH COUNTRY TO APPEAR ONLY ONCE.**

9 SELECT DISTINCT CountryOfOrigin FROM tblCoffee ORDER BY CountryOfOrigin . . . each country is only listed once.

**FIND ITEMS IN A GIVEN LIST WITHOUT USING COMPOUND OPERATORS (AND , OR)**

10 SELECT DISTINCT CountryOfOrigin FROM tblCoffee WHERE countryoforigin IN ("Ethiopia", "Limu", "Kenya", "Rwanda")  
ORDER BY CountryOfOrigin;

## Using a table of players as an example . . .

ID	Full Name	Nationality	Gender	DOB	Professional
1	Connie Rossetti	Greece	Female	17 Nov 2001	<input type="checkbox"/>
2	Boycey Wingar	China	Male	15 Jan 1978	<input checked="" type="checkbox"/>
3	Bert Laughrey	France	Female	22 Oct 1983	<input type="checkbox"/>
4	Flossi Couroy	China	Female	14 Dec 1982	<input checked="" type="checkbox"/>
5	Dian Ogborne	China	Female	30 Apr 1981	<input checked="" type="checkbox"/>
6	Tonia Gilardi	Indonesia	Female	09 Sep 1973	<input type="checkbox"/>
7	Consalve Lithcow	Russia	Male	04 Aug 1979	<input checked="" type="checkbox"/>
8	Sonnie Pessolt	Indonesia	Female	15 Oct 1986	<input type="checkbox"/>
9	Angelita Bertram	Indonesia	Female	14 Feb 1982	<input type="checkbox"/>
10	Reuben Oxbrow	Peru	Male	18 Oct 2000	<input type="checkbox"/>
11	Sayres Hansbury	Nigeria	Male	23 Nov 1974	<input type="checkbox"/>

**EXTRACT A PORTION OF A DATE - YEAR, MONTH, DAY FROM A FIELD THAT HAS THE DATA TYPE OF “Date”**

**“ [ ] “ brackets around Full Name because of the space in the middle of the field name**

**“AS” gives this extracted year value its own label ie “Year”**

```
10 SELECT [Full Name], YEAR(DOB) AS Year FROM tblPlayers ORDER BY [Full Name];
```

```
11 SELECT [Full Name], MONTH(DOB) AS Year FROM tblPlayers ORDER BY [Full Name];
```

```
12 SELECT [Full Name], DAY(DOB) AS Year FROM tblPlayers ORDER BY [Full Name];
```

**WE NEVER STORE DATA THAT WE CAN DETERMINE FROM OTHER FIELDS. Examples: ages, totals, averages, participation**

**DETERMINING AGE FROM DATE OF BIRTH AND SORTING FROM YOUNGEST TO OLDEST**

```
13 SELECT [Full Name], YEAR(NOW ( )) - YEAR(DOB) AS Age FROM tblPlayers ORDER BY YEAR(NOW ( )) - YEAR(DOB)
```

**EXTRACTING BIRTHDAYS THAT MATCH A CONDITION e.g. larger than January but smaller than June**

```
14 SELECT [Full Name], dob FROM tblPlayers WHERE month(dob) BETWEEN 1 AND 6;
```

**COMPARING AGES TO A PARTICULAR CONDITION e.g. smaller and equal to 20**

```
15 SELECT [Full Name], dob FROM tblPlayers WHERE YEAR(NOW()) - YEAR(dob) <= 20;
```

**DETERMINGING THE AVERAGE AGE IN YEARS, FROM DATE OF BIRTH**

```
16 SELECT AVG(YEAR(NOW()) - YEAR(dob)) AS [Average Age] FROM tblPlayers;
```

**DETERMINING THOSE NAMES THAT SO LONG THAT THEY WILL CREATE PROBLEMS WITH THE PRINTED MATERIAL.**

```
17 SELECT * FROM tblPlayers WHERE LEN([Full Name]) > 20 ORDER BY Nationality;
```

Using a table of machines that operate in a factory as an example . . .

ID	make	model	capacityPerHour	maxSize	YearOfManufa
1	Toshiba	FG-650	1000	200	2010
2	Toshiba	FG-850	2000	300	2014
3	Oshkosh	HAF-8000	2000	100	2014
4	Oshkosh	HGF-8500	2000	100	2008
5	Oshkosh	ZGF-9000	5000	300	2014
6	Big End	XC-450-A	1000	100	2010
7	Big End	XC-450-B	2000	200	2014
8	Big End	XC-450-C	3000	300	2014
9	Rawlinson	5650-AC	4000	400	2008
10	Rawlinson	5675-J	3000	500	2014
11	Rawlinson	7500-JC	2000	700	2016

#### FINDING A MACHINE THAT MATCHES A CONDITION

```
17 SELECT * FROM tblInventory WHERE make = "Rawlinson";
```

#### DETERMINING TOTAL CAPACITY PER HOUR OF ALL THE MACHINES IN THE FACTORY

```
18 SELECT SUM(capacityPerHour) AS [Total Capacity per Hour] from tblInventory;
```

#### MAKING A SERIAL NUMBER USING LETTERS FROM THE MAKE, MODEL AND FROM THE CAPACITY eg TF1000, TF2000, OH2000 etc

```
19 SELECT LEFT(make,1) & LEFT(model,1) & capacityPerHour AS [Serial Number] from tblInventory;
```

**LEFT and RIGHT allow us to give an index of how many characters we want. In this case we have only asked for one.**

Using a teacher's markbook as an example . . .

**Note:** We do not type in totals as they can be calculated from the figures already in the table. We say that they are derived fields i.e. derived from data already in the table.

SORT & FILTER		RECORDS				FIND	
ID	Maths	Eng	Science	IT	Art	Click	
1	46	63.5	78	50	87		
2	98	56	87	95	45		
3	45.5	76	87.5	87	87		
4	87	67	87	67.5	95		
5	93.5	59	54.5	62.5			

**FINDING VALUES THAT ARE MISSING FROM OUR TABLE USING NULL – WE USE “IS NULL” OR “IS NOT NULL”.**

```
20 SELECT * FROM tblStudents WHERE (maths is null) OR (eng is null) OR (science is null) OR (IT is null) OR (art is null);
```

**CALCULATING TOTALS AND AVERAGES FROM GIVEN SCORES – note comma after the wildcard**

```
21 SELECT *, ((maths + eng + science + it + art)/5) AS Average FROM tblStudents;
```

**EXTRACTING THOSE LEARNERS WHOSE AVERAGE IS HIGHER THAN 70%**

```
22 SELECT *, ((maths + eng + science + it + art)/5) AS Total FROM tblStudents WHERE ((maths + eng + science + it + art)/5) > 70;
```

**WHEN WE USE AN AGGREGATE FUNCTION LIKE SUM BUT WHAT TO SEE OTHER FIELDS AS WELL – WE USE GROUP BY**

```
23 SELECT ID, SUM (((maths + eng + science + it + art))) AS Total FROM tblStudents GROUP BY ID;
```