

SQL Summary – Version5

SELECT <field(s)>

FROM <table(s)>

WHERE condition(s)

GROUP BY expression

HAVING condition – only works with GROUP BY

ORDER BY expression

=====

SELECT ALL fields

1 SELECT * FROM tablename

SELECT ... FROM ... SELECT selects fields (only). Also the wildcard

2 SELECT * FROM tablename ... "*" wildcard selects everything – all fields.

3 SELECT * FROM tblName WHERE LastName LIKE "W*"

ORDER BY – ASC or DESC. ASC is default – smallest first, largest last. A first, Z last.

4 SELECT * FROM tblName ORDER BY LastName

5 SELECT * FROM tblName ORDER BY price DESC ... the highest price is first

6 SELECT * FROM tblInventory ORDER BY price, make ... cheapest to most expensive in alphabetical order by make

THE BEST ... THE 5 BEST ... THE WORST ... THE HIGHEST ... THE MOST (EXPENSIVE) (CHEAPEST) ... EARLIEST ... LATEST

7 SELECT **TOP 1** * FROM tblName ORDER BY ASC / SELECT **TOP 5** * FROM ORDER BY DESC.

8 SELECT TOP 20 * FROM tblPrintLogs ORDER BY TotalColourPages DESC

9 SELECT TOP 1 surname FROM tblPrintLogs ORDER BY HOUR(date) DESC

10 SELECT TOP 1 lastName, firstName FROM tblPrintLogs WHERE email LIKE “*student*” ORDER BY cost DESC

SELECT ONLY CERTAIN FIELDS AND THEN SORT BY A SPECIFIC FIELD

11 SELECT field1, field2, field3, fieldN FROM tblName ORDER BY Age

SELECT VALUES THAT ARE UNIQUE

12 SELECT DISTINCT fields FROM tblName ORDER BY eg SELECT DISTINCT LastName FROM tblStudents

PERFORM ARITHMETIC ON THE SELECTED FIELDS AND THEN NAME THIS “NEW” FIELD (called an Alias)

13 SELECT fields calculation FROM ORDER BY eg SELECT Name, (Math + Science + IT) AS Total FROM tblStudents

USE ARITHMETIC FUNCTION TO FORMAT A RESULT eg rounding to 1 decimal place

14 SELECT fields arithmetic function(calculation) eg SELECT Name ROUND((Math + IT) /2, 1) AS Average FROM tblStudents

15 SELECT fields calculations (MOD) AS alias eg SELECT totalPages MOD 4 AS Total

STRING MANIPULATION TO EXTRACT A PORTION OF A LONGER STRING eg finding a person’s initial

16 SELECT string manipulation (field) & field & field eg SELECT LEFT(firstName,1)

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

17 SELECT date or time(field) ... eg SELECT YEAR(DOB) SELECT MONTH(DOB)

PERFORMING AGGREGATE FUNCTIONS ON ALL THE VALUES IN A PARTICULAR FIELD. YIELDS A SINGLE VALUE.

18 SELECT aggregate function(field) ... eg SELECT MIN(SizeKB) ... SELECT AVG(TotalPages) ... SELECT SUM(Cost)

WHERE - COMPARING VALUES IN A PARTICULAR FIELD TO A PARTICULAR CONDITION eg larger than 5

19 SELECT fields WHERE condition = < > <= >= <> ... eg SELECT surname FROM PrintLogs WHERE TotalPages > 5 ...

20 SELECT Name FROM Table WHERE Science > (SELECT AVG (Science) FROM Table) ...

21 SELECT Name FROM tblStudents WHERE Address1 IS NULL

22 SELECT * FROM tblName WHERE ModelNumber LIKE "XC-450?" – wildcard ? for a single character

DETERMINING AGE AND DISPLAYING AGE FROM DATE OF BIRTH

23 SELECT Name, YEAR(NOW ()) – YEAR(DOB) AS Age

COMPARING AGES TO A PARTICULAR CONDITION eg larger and equal to 30

24 SELECT fields FROM WHERE calculation condition ... eg SELECT fields WHERE YEAR(NOW()) – YEAR(DOB) <= 30

COMPARING AGES TO MORE THAN ONE CONDITION eg larger than March but smaller than June

25 SELECT fields WHERE condition AND condition ... eg SELECT fields WHERE MONTH(DOB) BETWEEN 3 AND 6.

DETERMINING THE AVERAGE AGE FROM DATE OF BIRTH

26 SELECT Name, AVG(NOW() – YEAR(DOB)) AS AveAge

PERFORMING A SIMPLE CALCULATION ON A NUMBER OF FIELDS IN THE SAME RECORD

27 SELECT (TotalPages * Copies) ... (Maths + Science + IT) ... (LeaveDaysPerYear – DaysTakenThisYear)

USING GROUP BY and HAVING with aggregate functions (aggregate function only provide one result unless . . .)

GROUP BY – The result of an aggregate function can offer more information if grouped into categories satisfying a condition

28 SELECT continent, COUNT(countryName) FROM tblWorld GROUP BY continent . . . for number of countries on each continent

29 SELECT continent, COUNT(countryName) FROM tblWorld WHERE population > 20000000 GROUP BY continent

30 SELECT continent, SUM(population) FROM tblWorld GROUP BY continent . . . the total population on each continent

31 SELECT continent, SUM(population) FROM tblWorld GROUP BY continent HAVING SUM(population) > 50000000

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

Comparison operators: < > >= <= <> (not equal to)

Random numbers: RND(Upper bound – Lower bound) + Lower bound

String manipulation: LEFT(x), RIGHT(x), MID(x, y), LEN, & - concatenation operator in Ms Access

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 (value) eg it adds all the values in a field when you SUM. They do not return any details from a specific record. Example: If you need maximum and minimum values *with details* use ORDER BY.

Most useful when used in conjunction with GROUP BY

COUNT does not count a record that has a NULL value in the specified field. The other aggregate functions ignore NULL values eg SUM

WHERE TotalPages > AVG(TotalPages) . . . *does not work*. **You cannot compare a value to an aggregate.**

WHERE TotalPages > (SELECT AVG(TotalPages) FROM PrintLogs) . . . this works because each statement presents a value.

SELECT make, model, price FROM tblInventory WHERE price < (SELECT AVG(price) FROM tblInventory.) See 14 for another example.

Each aggregate function must have its **own** SELECT statement so that it presents its own value for comparison

WHERE (SELECT AVG(field1) FROM Table1) > WHERE (SELECT AVG(field2) FROM Table1).

Compound conditions: NOT, AND, OR

WHERE town = "Johannesburg" AND maritalstatus = 1 AND gender = 1

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

SELECT * FROM tblDetails WHERE town IN ("Johannesburg", "Pretoria", "Midrand")

SELECT * FROM tblDetails WHERE town NOT IN ("Johannesburg", "Pretoria", "Midrand")

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

SQL Summary – Queries that alter data in a table (insert records, delete records or edit existing records) (table is not altered)

32 INSERT – adds a new record to a table and populates all the fields (not when autonumber is the primary key)

INSERT INTO tablename VALUES (field1Data, field2Data, field2Data) – no field names.

NOTE: The VALUES, the order of the values, and the datatypes match the table structure exactly

33 INSERT - adds a new record, specified fields ... (useful when the first field is autonumber)

INSERT INTO tablename (fieldTitle1, fieldTitle2, fieldTitle3) VALUES (field1Data, field2Data, field2Data)

Eg INSERT INTO tblname (name, DOB, gender, grade, boarder) VALUES ('Lynn', #02 Feb 2000#, 'F', 10, false)

34 UPDATE – all ... (the whole table, and all its records are given a new value e.g. the school gets a new name – everybody is affected.

UPDATE tablename SET field1 = value1, field2 = value2, fieldN = valueN

35 UPDATE – updates a record that matches a condition

UPDATE tablename SET field1 = value1, field2 = value2, fieldN = valueN WHERE condition

E.g. UPDATE PrintLogs SET FirstName = “Henrietta” WHERE Surname = “Bates” AND FirstName = “Henry”.

36 DELETE – all . . . (Deletes all the records in the table and cannot be undone in Ms Access. The table structure is not affected)

DELETE * FROM tablename

37 DELETE – those that match a condition . . . (NOTE: This delete SQL command cannot be undone Ms Access)

DELETE FROM tablename WHERE fieldname = value

E.g. DELETE FROM tblStudents WHERE studentID = 38 (best to delete using the primary key value, not the person’s name)

More examples – SELECT

1. SELECT *
2. SELECT name, region
3. SELECT name, area/population . . . (area divided by population which gives the population density)
4. SELECT ROUND(area/population, 2) . . . as above rounded to 2 decimal places
5. SELECT LENGTH(name)
6. SELECT name, LEFT(name,1)
7. SELECT name FROM tblWorldStats WHERE population > (SELECT population FROM tblWorldStats WHERE name = “Russia”)
 - a. Note: A query within a query – the second SELECT must only return one value or the comparison operator cannot work.
8. SELECT INT(genderMale / totalEnrolement * 100) . . . Percentage of male students enrolled in a college rounded down.
9. SELECT ROUND(genderMale / totalEnrolement * 100, 2) . . . Percentage of male students enrolled in a college rounded to 2 decimal places
10. SELECT ROUND(RND(SizeKb) * 5,0) . . . whole random numbers from 1 to 5 inclusive.

11. SELECT LastName & " " & LEFT(firstName,1) AS LastNameInit . . . last name concatenated with initial.

More examples – WHERE

1. WHERE area = 1000000
2. WHERE country = 'Germany'
3. WHERE country IN ('United Kingdom', 'Europe', 'Asia')
4. WHERE MONTH(DOB) IN (1,2,3) . . . finds people born in the first quarter of the year
5. WHERE name LIKE "A*"
6. WHERE name LIKE "A*" OR "E*"
7. WHERE length(name) > 10 AND region = "Europe"
8. WHERE area < 500000 AND population > 1000000
9. WHERE area BETWEEN 1000000 AND 2000000
10. WHERE nationality = 'England' AND goalsScored BETWEEN 40 AND 50
11. WHERE subject = 'English' AND YEAR(publication) BETWEEN 2000 AND 2015
12. WHERE LastName BETWEEN 'A' AND 'M'
13. WHERE DOB BETWEEN #01/01/2010# AND #31/12/2010#
14. WHERE MONTH(DOB) = 12 AND DAY(DOB) = 25 finds people born on Christmas Day.
15. WHERE name NOT LIKE " * * " . . . space in the middle. Names made of two separate words would not be selected.

RESOURCE:

www.sqlzoo.net - useful teach, example and quiz website. Note: Does not focus on Ms Access SQL implementation – small differences.