Part I. Inside Table

1. [SELECT ... FROM ... ;] <注意每句话都要加分号，大写的是keywords>

- SELECT 【col_name】 FROM 【data_table_name】; [Note: 一个dataset里面有几个table(tab)，如Excel]
- SELECT 【col_name_1, col_name_2, col_name_3】 FROM 【data_table_name】; [SELECT 后面只能跟逗号形式不能用AND]
- SELECT * FROM 【data_table_name】; [全选，显示所有细节行列]
- SELECT * FROM 【data_table_name】 LIMIT 10; [选10个]
- SELECT 【'string'】 AS result; [will return the "string" contents]
- SELECT distinct【col_name】 FROM 【data_table_name】; [only return the unique ones]
- SELECT COUNT(*) FROM 【data_table_name】; [return the number of rows of the data table (including missing values)]
- SELECT COUNT(label) FROM 【data_table_name】; [count all the NON-MISSING values of a single column]
- SELECT COUNT(distinct 【col_name】) FROM 【data_table_name】; [count the unique ones]

Note:

- column – field

2. Filtering Results

- = equal [注意SQL里不是==]
  - <> or != not equal [NOTE: <> is ISO standard]
  - \< less than
  - > more than
  - \<= less equal
  - >= greater equal
- SELECT ... FROM ... WHERE (label);
- SELECT COUNT (*) FROM ... WHERE (label) < (number); [count, 范围全选，missing values 不算]
- SELECT 【col_name】 FROM 【data_table_name】 WHERE 【col_name】 = 'string'; [注意SQL只能单引号！！]
- SELECT 【col_name】 FROM 【data_table_name】 WHERE 【condition】 AND 【condition】 AND 【condition】; [可以加无限个AND]
- SELECT 【col_name】 FROM 【data_table_name】 WHERE (... OR ...) AND (... OR ...); [AND和OR都有时注意加括号]
- SELECT 【col_name】 FROM 【data_table_name】 WHERE 【col_name】BETWEEN ... AND ...; [Note: BETWEEN is inclusive]
- SELECT 【col_name_1, col_name_2, ...】 FROM 【data_table_name】 WHERE 【col_name】 IN (condition_1, condition_2, condition_3, ...); [注意不是range是match]
- SELECT 【col_name_1】 AS 【new_col_name】 FROM 【data_table_name_1】 WHERE 【col_name_1】 NOT IN (SELECT 【col_name_2】 FROM 【data_table_name_2】); [SELECT套用，注意XX不在XX list里面要重新用SELECT] [如果是unique的column name可以不用【.】不然要用以分清]
- SELECT 【col_name】 FROM 【data_table_name】 WHERE 【col_name】 IS NULL / IS NOT NULL;
- SELECT COUNT(*) FROM 【data_table_name】 WHERE 【col_name】 IS NULL / IS NOT NULL; [count the missing values of a specific column]
- SELECT 【col_name】 FROM 【data_table_name】 WHERE 【col*name】 LIKE / NOT LIKE 'string'* / 'string%' / '_string%'; [string match，前后都可加]

Note:

- WHERE comes after FROM
- Clause: WHERE; Operator: LIKE
- Wildcard: 【%】 - matches one or more character(s); 【_】 - matches one character
- In SQL, NULL represents missing values or unknown values.
- Need to specify the column for every "OR" condition
  For example,
  - SELECT ... FROM ... WHERE year = 2000 OR 1990
    is WRONG
  - SELECT ... FROM ... WHERE year = 2000 OR year = 1990
    is CORRECT

3. Aggregate Functions

- SELECT function(【col_name】) FROM
- AVG() [average]
- MAX() / MIN()
- SUM()
- SELECT (【+】/【-】/【*】/【/】) [注意除法只取商整数]
  - 10/3 = 3
  - 10/3.0 = 1.33333...
  - 10%3 = 1
  - 10%3.0 = 1.0
  - 5/2.0 = 2.50000...
  - 45*100.0 = 4500.0
  - 45*100.00 = 4500.00
- SELECT (【+】/【-】/【*】/【/】) AS RESULT
- SELECT 【...】 AS 【new_col_name_1】, 【...】 AS 【new_col_name_2】 FROM 【data_table_name】
- SELECT COUNT(【col_name】) * 100.0 / count(【data_table_name】) FROM 【data_table_name】; [calculate the percentage]

4. Sorting and Grouping

- SELECT 【col_name】 FROM 【data_table_name】 ORDER BY 【col_name】【keyword, i.e. DESC】; [if there is any keyword]
- SELECT 【col_name】 FROM 【data_table_name】 WHERE 【condition】 ORDER BY 【col_name_1】, 【col_name_2】, ...;
- SELECT 【col_name】 FROM 【data_table_name】 GROUP BY 【col_name】;
- SELECT count(*) FROM 【data_table_name】 GROUP BY 【col_name】; [does NOT show the group column, change to SELECT 【col_name】, count(*) to show two columns]
- SELECT 【col_name_1】, 【col_name_2】, ... FROM 【data_table_name】 GROUP BY 【col_name_1】, 【col_name_2】, ... ORDER BY 【col_name_1】, 【col_name_2】, ...;
- SELECT 【col_name_1】, ... FROM 【data_table_name】 GROUP BY 【col_name_1】, ... ORDER BY 【col_name_1】, ... HAVING 【condition】; [HAVING is similar to WHERE, WHERE cannot be used for functions but HAVING can] [用GROUP BY即使是空集也会显示那一列]
- SELECT 【col_name_1】, ... FROM 【data_table_name】 GROUP BY 【col_name_1】, ... ORDER BY 【col_name_1】, ... HAVING 【condition】 LIMIT 【#number】; [show #number of rows of results]
- SELECT 【col_name_1】, 【col_name_category】... FROM 【data_table_name】 GROUP BY 【col_name_1】, ... ORDER BY 【col_name_1】, ... HAVING COUNT(DISTINCT(【col_name_1】)) 【condition】; [以类别归类统计数量]

Note:

- Keywords: (1) DESC – descending, (2) LIMIT
- ORDER: by alphabet
- GROUP BY 放在HAVING前面！ORDER BY 放在 WHERE 后面！
- Aggregate function CANNOT be used in WHERE, need to use another clause: HAVING

5. Alter Table

- ##Add a new column##
  - ALTER TABLE【data_table_name】;
  - ADD 【new_col_name】 【datatype】; [Note: nothing in the new column]
- ##Drop an existing column##
  - ALTER TABLE【data_table_name】 ;
  - DROP COLUMN 【col_name】;
- ##Modify the datatype of an existing column##
  - ALTER TABLE【data_table_name】;
  - MODIFY COLUMN 【col_name】 【datatype】

6. Update Tables

- UPDATE 【data_table_name】 SET 【col_name】 = IF(【col_name】 = 'string', 'name_1', 'name_2'); [the same as if else statement, binary case]
  or
  UPDATE 【data_table_name】 SET 【col_name】 = CASE 【col_name】 WHEN 'string_1' THEN 'name_1' ELSE 'name_2') END; [the same as if else statement, binary case]

7. Delete

- DELETE 【new_table_name_A】 FROM 【table_name_1】 AS 【new_table_name_A】, 【table_name_1】 AS 【new_table_name_B】WHERE 【A.col_name_1】 = 【B.col_name_1】 AND 【A.col_name_2】 > 【B.col_name_2】; [delete duplicated columns]

8. Skip Rows

- SELECT 【col_name】 FROM 【data_table_name】 OFFSET 【# number】; [跳过#行]

9. Null

- SELECT IFNULL((SELECT 【col_name】 FROM 【data_table_name】), NULL) AS 【new_col_name】; [如果有return里面的SELECT的内容，不然return NULL]
- SELECT(SELECT 【col_name】 FROM 【data_table_name】) AS【new_col_name】; [可省略为两个SELECT，AS放最后就好]

10. Date

- SELECT 【A.col_name】 FROM 【table_name_1】 AS A, 【table_name_1】 AS B WHERE DATEDIFF(【date_col_name_1】, 【date_col_name_1】) = 1 AND 【condition】; [date difference = 1]

11. Rank

- RANK() OVER 【col_name】;
- DENSE_RANK() OVER 【col_name】;