Table creation code

**CREATE** **TABLE** Warehouses (

Code INTEGER **NOT** **NULL**,

**Location** VARCHAR(255) **NOT** **NULL** ,

Capacity INTEGER **NOT** **NULL**,

**PRIMARY** **KEY** (Code)

);

**CREATE** **TABLE** Boxes (

Code VARCHAR(255) **NOT** **NULL**,

Contents VARCHAR(255) **NOT** **NULL** ,

Value REAL **NOT** **NULL** ,

Warehouse INTEGER **NOT** **NULL**,

**PRIMARY** **KEY** (Code),

**FOREIGN** **KEY** (Warehouse) **REFERENCES** Warehouses(Code)

) ENGINE=INNODB;

Sample dataset

**INSERT** **INTO** Warehouses(Code,**Location**,Capacity) **VALUES**(1,'Chicago',3);

**INSERT** **INTO** Warehouses(Code,**Location**,Capacity) **VALUES**(2,'Chicago',4);

**INSERT** **INTO** Warehouses(Code,**Location**,Capacity) **VALUES**(3,'New York',7);

**INSERT** **INTO** Warehouses(Code,**Location**,Capacity) **VALUES**(4,'Los Angeles',2);

**INSERT** **INTO** Warehouses(Code,**Location**,Capacity) **VALUES**(5,'San Francisco',8);

**INSERT** **INTO** Boxes(Code,Contents,Value,Warehouse) **VALUES**('0MN7','Rocks',180,3);

**INSERT** **INTO** Boxes(Code,Contents,Value,Warehouse) **VALUES**('4H8P','Rocks',250,1);

**INSERT** **INTO** Boxes(Code,Contents,Value,Warehouse) **VALUES**('4RT3','Scissors',190,4);

**INSERT** **INTO** Boxes(Code,Contents,Value,Warehouse) **VALUES**('7G3H','Rocks',200,1);

**INSERT** **INTO** Boxes(Code,Contents,Value,Warehouse) **VALUES**('8JN6','Papers',75,1);

**INSERT** **INTO** Boxes(Code,Contents,Value,Warehouse) **VALUES**('8Y6U','Papers',50,3);

**INSERT** **INTO** Boxes(Code,Contents,Value,Warehouse) **VALUES**('9J6F','Papers',175,2);

**INSERT** **INTO** Boxes(Code,Contents,Value,Warehouse) **VALUES**('LL08','Rocks',140,4);

**INSERT** **INTO** Boxes(Code,Contents,Value,Warehouse) **VALUES**('P0H6','Scissors',125,1);

**INSERT** **INTO** Boxes(Code,Contents,Value,Warehouse) **VALUES**('P2T6','Scissors',150,2);

**INSERT** **INTO** Boxes(Code,Contents,Value,Warehouse) **VALUES**('TU55','Papers',90,5);

Exercises

1. Select all warehouses.

https://upload.wikimedia.org/wikipedia/commons/1/10/MediaWiki_Vector_skin_action_arrow.pngClick to see solution

**SELECT** \* **FROM** Warehouses;

2. Select all boxes with a value larger than $150.

https://upload.wikimedia.org/wikipedia/commons/1/10/MediaWiki_Vector_skin_action_arrow.pngClick to see solution

**SELECT** \* **FROM** Boxes

**WHERE** Value > 150;

3. Select all distinct contents in all the boxes.

https://upload.wikimedia.org/wikipedia/commons/1/10/MediaWiki_Vector_skin_action_arrow.pngClick to see solution

**SELECT** **DISTINCT** Contents

**FROM** Boxes;

4. Select the average value of all the boxes.

https://upload.wikimedia.org/wikipedia/commons/1/10/MediaWiki_Vector_skin_action_arrow.pngClick to see solution

**SELECT** **AVG**(Value)

**FROM** Boxes;

5. Select the warehouse code and the average value of the boxes in each warehouse.

https://upload.wikimedia.org/wikipedia/commons/1/10/MediaWiki_Vector_skin_action_arrow.pngClick to see solution

**SELECT** Warehouse, **AVG**(Value)

**FROM** Boxes

**GROUP** **BY** Warehouse;

6. Same as previous exercise, but select only those warehouses where the average value of the boxes is greater than 150.

https://upload.wikimedia.org/wikipedia/commons/1/10/MediaWiki_Vector_skin_action_arrow.pngClick to see solution

**SELECT** Warehouse, **AVG**(Value)

**FROM** Boxes

**GROUP** **BY** Warehouse

**HAVING** **AVG**(Value) > 150;

7. Select the code of each box, along with the name of the city the box is located in.

https://upload.wikimedia.org/wikipedia/commons/1/10/MediaWiki_Vector_skin_action_arrow.pngClick to see solution

**SELECT** Boxes.Code, **Location**

**FROM** Warehouses **INNER** **JOIN** Boxes

**ON** Warehouses.Code = Boxes.Warehouse;

8. Select the warehouse codes, along with the number of boxes in each warehouse. Optionally, take into account that some warehouses are empty (i.e., the box count should show up as zero, instead of omitting the warehouse from the result).

https://upload.wikimedia.org/wikipedia/commons/1/10/MediaWiki_Vector_skin_action_arrow.pngClick to see solution

*/\* Not taking into account empty warehouses \*/*

**SELECT** Warehouse, **COUNT**(\*)

**FROM** Boxes

**GROUP** **BY** Warehouse;

*/\* Taking into account empty warehouses \*/*

**SELECT** Warehouses.Code, **COUNT**(Boxes.Code)

**FROM** Warehouses **LEFT** **JOIN** Boxes

**ON** Warehouses.Code = Boxes.Warehouse

**GROUP** **BY** Warehouses.Code;

9. Select the codes of all warehouses that are saturated (a warehouse is saturated if the number of boxes in it is larger than the warehouse's capacity).

https://upload.wikimedia.org/wikipedia/commons/1/10/MediaWiki_Vector_skin_action_arrow.pngClick to see solution

**SELECT** Code

**FROM** Warehouses

**WHERE** Capacity <

(

**SELECT** **COUNT**(\*)

**FROM** Boxes

**WHERE** Warehouse = Warehouses.Code

);

*/\* Alternate method not involving nested statements \*/*

**SELECT** Warehouses.Code

**FROM** Warehouses **JOIN** Boxes **ON** Warehouses.Code = Boxes.Warehouse

**GROUP** **BY** Warehouses.code, Warehouses.Capacity

**HAVING** **Count**(Boxes.code) > Warehouses.Capacity

10. Select the codes of all the boxes located in Chicago.

https://upload.wikimedia.org/wikipedia/commons/1/10/MediaWiki_Vector_skin_action_arrow.pngClick to see solution

*/\* Without subqueries \*/*

**SELECT** Boxes.Code

**FROM** Warehouses **RIGHT** **JOIN** Boxes

**ON** Warehouses.Code = Boxes.Warehouse

**WHERE** **Location** = 'Chicago';

*/\* With a subquery \*/*

**SELECT** Code

**FROM** Boxes

**WHERE** Warehouse **IN**

(

**SELECT** Code

**FROM** Warehouses

**WHERE** **Location** = 'Chicago'

);

11. Create a new warehouse in New York with a capacity for 3 boxes.

https://upload.wikimedia.org/wikipedia/commons/1/10/MediaWiki_Vector_skin_action_arrow.pngClick to see solution

**INSERT**

**INTO** Warehouses

(**Location**,Capacity)

**VALUES** ('New York',3);

12. Create a new box, with code "H5RT", containing "Papers" with a value of $200, and located in warehouse 2.

https://upload.wikimedia.org/wikipedia/commons/1/10/MediaWiki_Vector_skin_action_arrow.pngClick to see solution

**INSERT** **INTO** Boxes

**VALUES**('H5RT','Papers',200,2);

13. Reduce the value of all boxes by 15%.

https://upload.wikimedia.org/wikipedia/commons/1/10/MediaWiki_Vector_skin_action_arrow.pngClick to see solution

**UPDATE** Boxes **SET** Value = Value \* 0.85;

14. Apply a 20% value reduction to boxes with a value larger than the average value of all the boxes.

Attention: this solution doesn't work with MySQL 5.0.67: *ERROR 1093 (HY000): You can't specify target table 'Boxes' for update in FROM clause*

https://upload.wikimedia.org/wikipedia/commons/1/10/MediaWiki_Vector_skin_action_arrow.pngClick to see solution

**UPDATE** Boxes **SET** Value = Value \* 0.80

**WHERE** Value > (**SELECT** **AVG**(Value) **FROM** (**SELECT** \* **FROM** Boxes) **AS** X);

15. Remove all boxes with a value lower than $100.

https://upload.wikimedia.org/wikipedia/commons/1/10/MediaWiki_Vector_skin_action_arrow.pngClick to see solution

**DELETE** **FROM** Boxes **WHERE** Value < 100;

16. Remove all boxes from saturated warehouses.

Attention: this solution doesn't work with [MySQL](https://en.wikibooks.org/wiki/MySQL) 5.0.67: *ERROR 1093 (HY000): You can't specify target table 'Boxes' for update in FROM clause*

**DELETE** **FROM** Boxes **WHERE** Warehouse **IN**

(

**SELECT** Code

**FROM** Warehouses

**WHERE** Capacity <

(

**SELECT** **COUNT**(\*)

**FROM** Boxes

**WHERE** Warehouse = Warehouses.Code

)

);