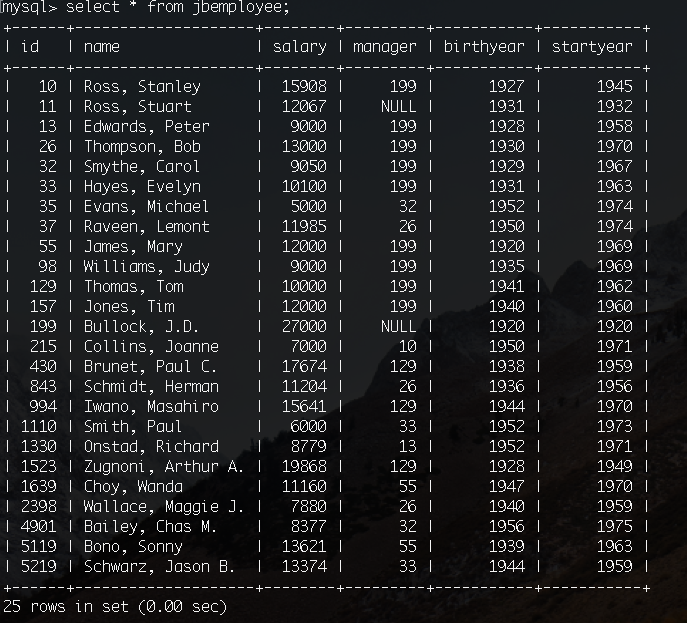
lab1\_report

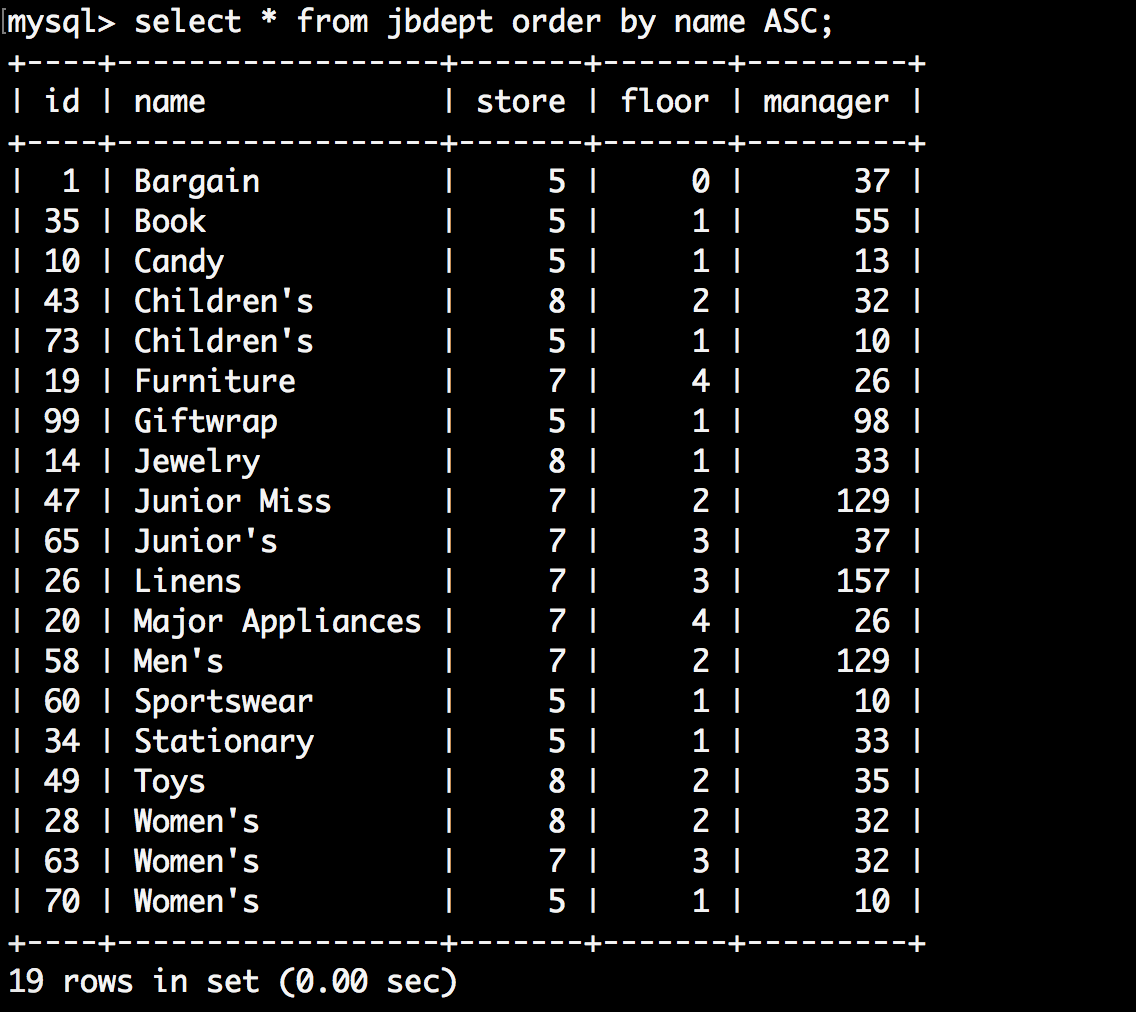
Rabnawaz jansher (rash696) & Saman Zahid (samza595)

# Lab 1 SQL-Queries and Views

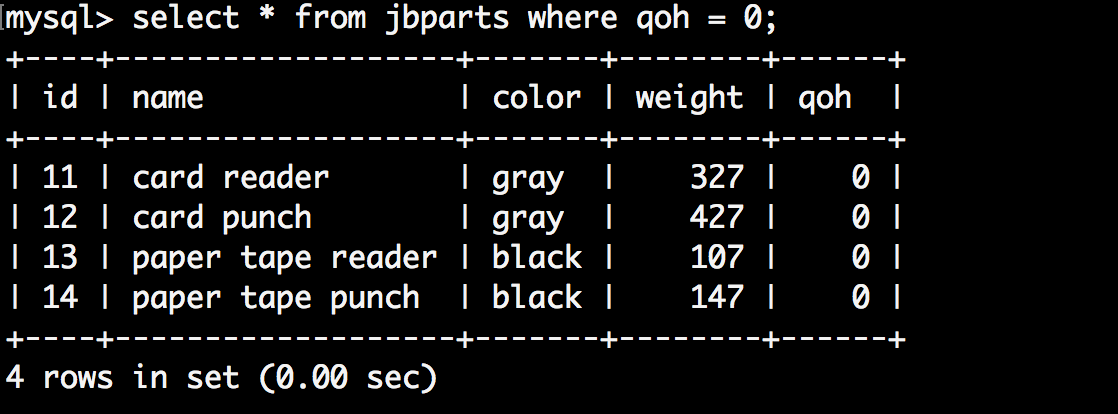
## 1) List all employees, i.e. all tuples in the jbemployee relation.



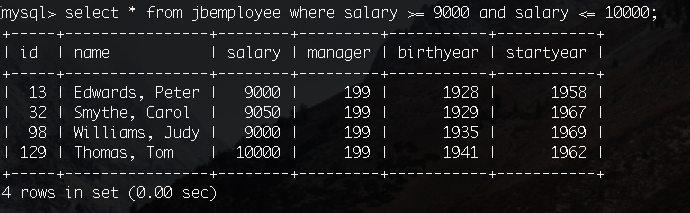
## 2) List the name of all departments in alphabetical order. Note: by “name” we mean the name attribute for all tuples in the jbdept relation.



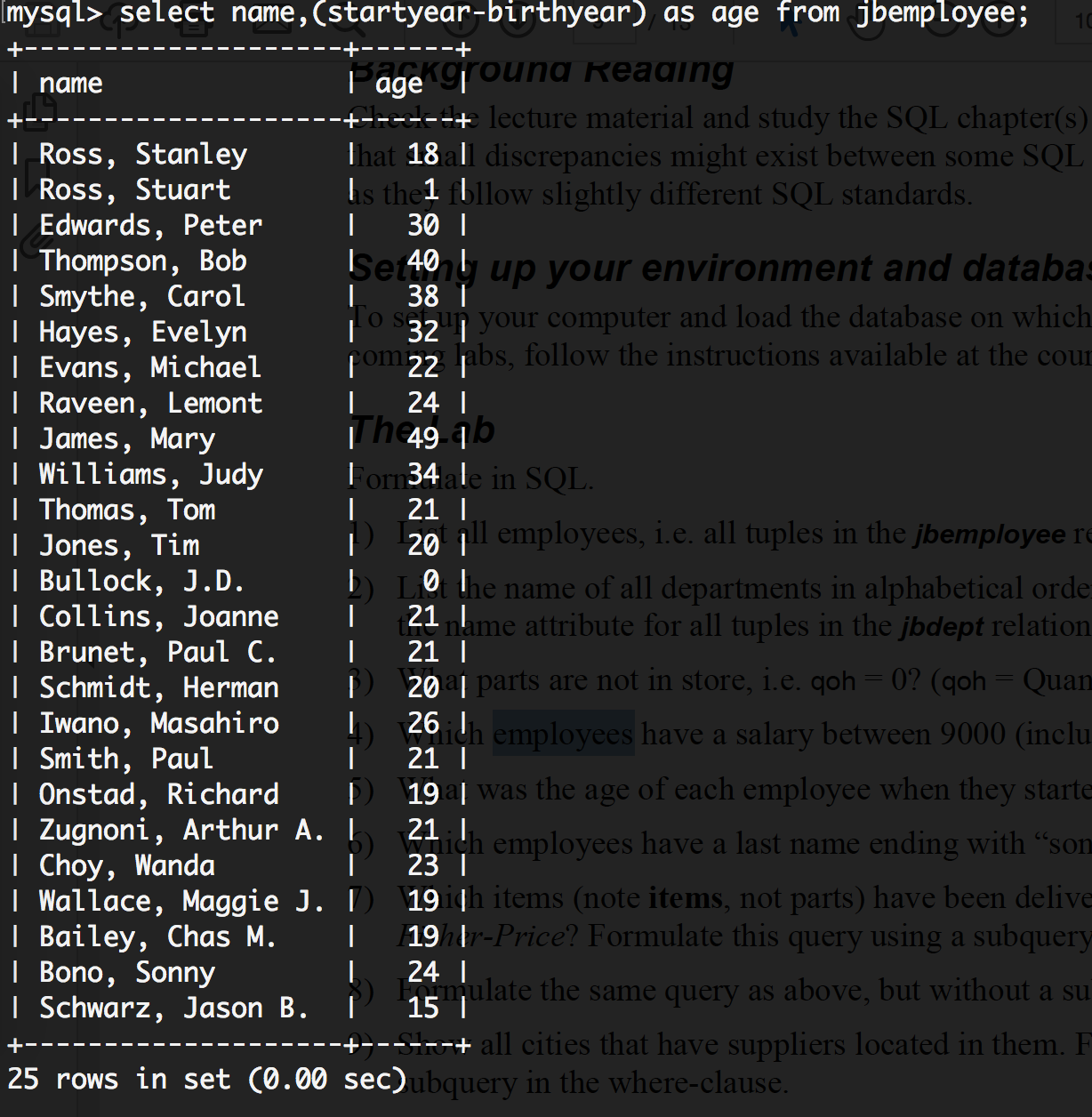
## 3) What parts are not in store, i.e. qoh = 0? (qoh = Quantity On Hand)



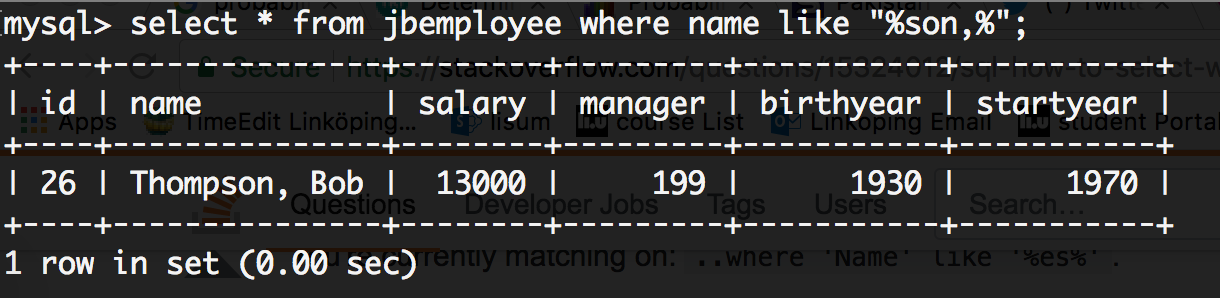
## 4) Which employees have a salary between 9000 (included) and 10000 (included)?



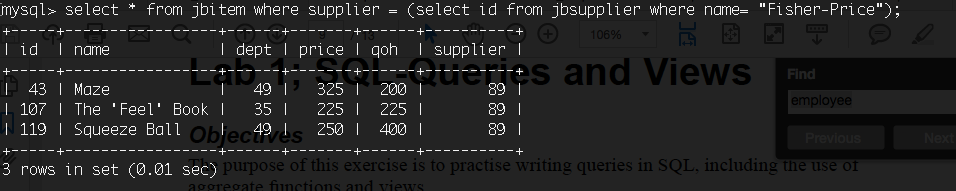
## 5) What was the age of each employee when they started working (startyear)?



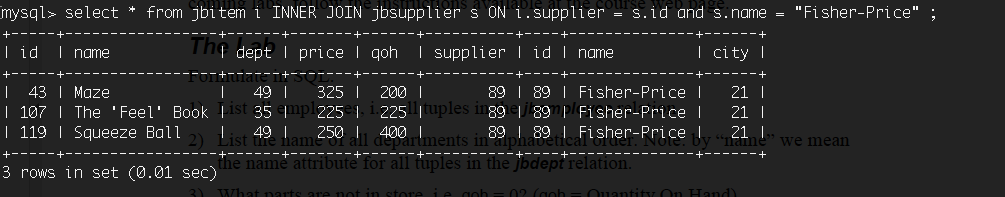
## 6) Which employees have a last name ending with “son”?



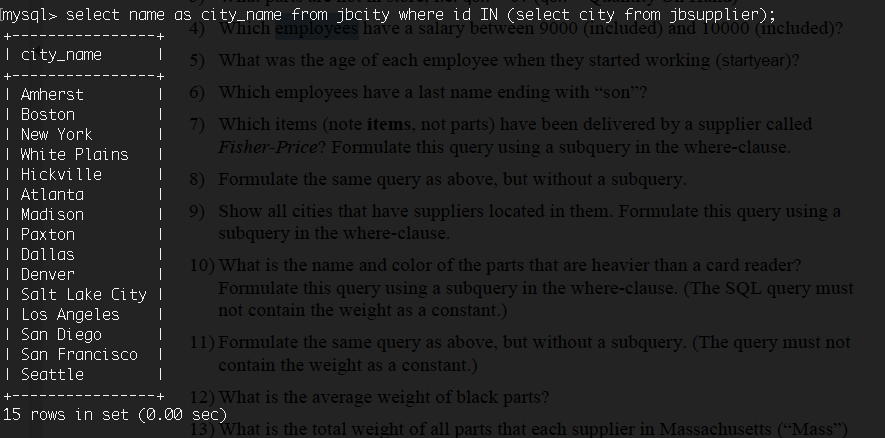
## 7) Which items (note items, not parts) have been delivered by a supplier called Fisher-Price? Formulate this query using a subquery in the where-clause



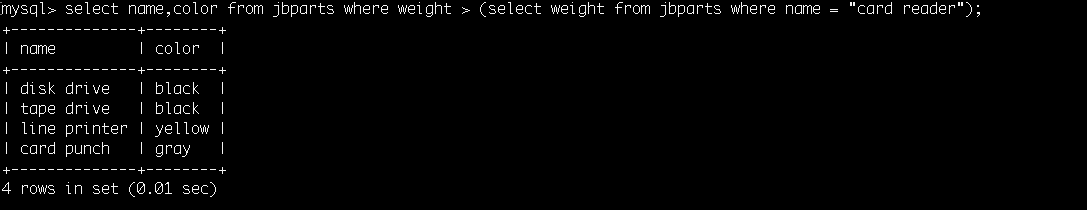
## 8) Formulate the same query as above, but without a subquery.



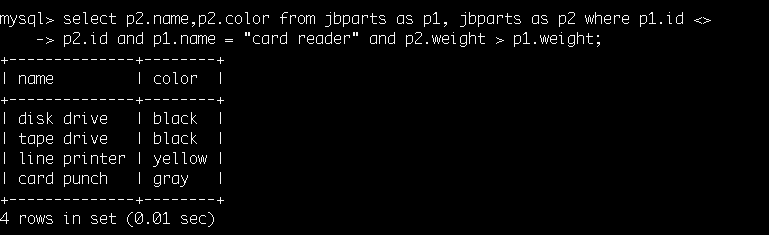
## 9) Show all cities that have suppliers located in them. Formulate this query using a subquery in the where-clause.



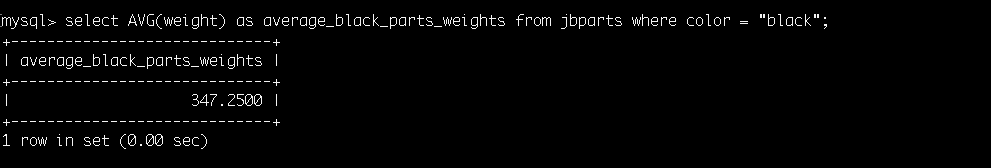
## 10) What is the name and color of the parts that are heavier than a card reader? Formulate this query using a subquery in the where-clause. (The SQL query must not contain the weight as a constant.)



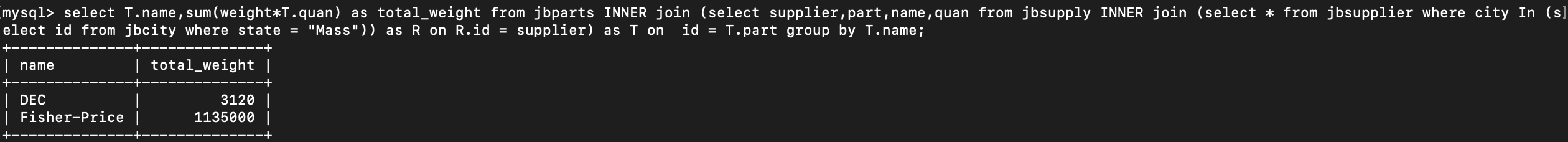
## 11) Formulate the same query as above, but without a subquery. (The query must not contain the weight as a constant.)



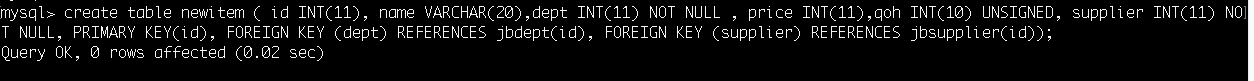
## 12) What is the average weight of black parts?

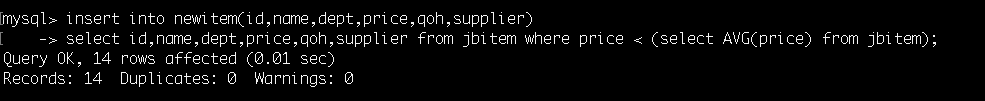


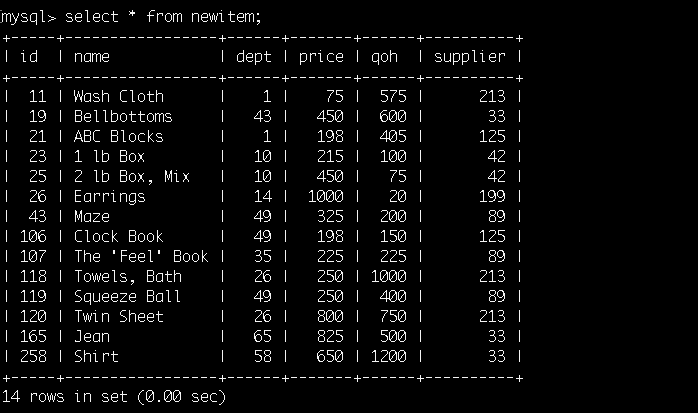
## 13) What is the total weight of all parts that each supplier in Massachusetts (“Mass”) has delivered? Retrieve the name and the total weight for each of these suppliers. Do not forget to take the quantity of delivered parts into account. Note that one row should be returned for each supplier.



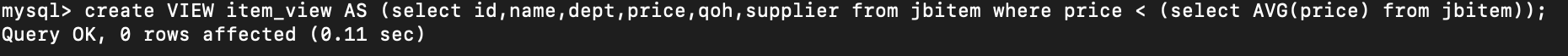
## 14) Create a new relation (a table), with the same attributes as the table items using the CREATE TABLE syntax where you define every attribute explicitly (i.e. not as a copy of another table). Then fill the table with all items that cost less than the average price for items. Remember to define primary and foreign keys in your table!

Create a table-newitem 

Insert into Table \_\_newitem\_\_ 



## 15) Create a view that contains the items that cost less than the average price for items.



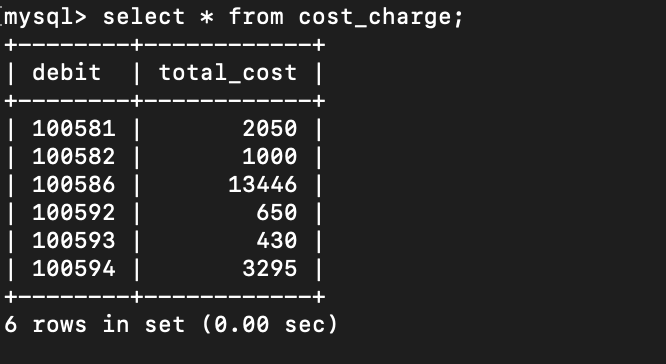
## 16) What is the difference between a table and a view? One is static and the other is dynamic. Which is which and what do we mean by static respectively dynamic.

View is kind a virtual table based on result of some query or muliple queries , which contains the same columns and rows as the number of rows and columns in resultant query/table . View is dynamic ,because when we update a value in original table it also update and modify the value in view table which is act as dynamic virtual table/relation.

On other, table is static , based on result of some query. if we update in a original table in doesnot update in new table(which is clone of original table). so table is static.

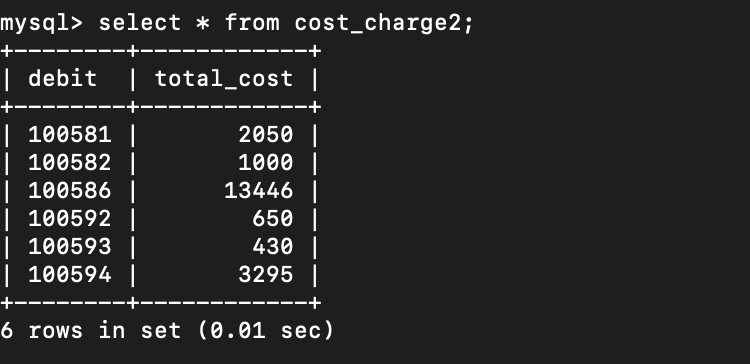
## 17) Create a view, using only the implicit join notation, i.e. only use where statements but no inner join, right join or left join statements, that calculates the total cost of each debit, by considering price and quantity of each bought item. (To be used for charging customer accounts). The view should contain the sale identifier (debit) and total cost.





## 18) Do the same as in (17), using only the explicit join notation, i.e. using only left, right or inner joins but no where statement. Motivate why you use the join you do (left, right or inner), and why this is the correct one (unlike the others).

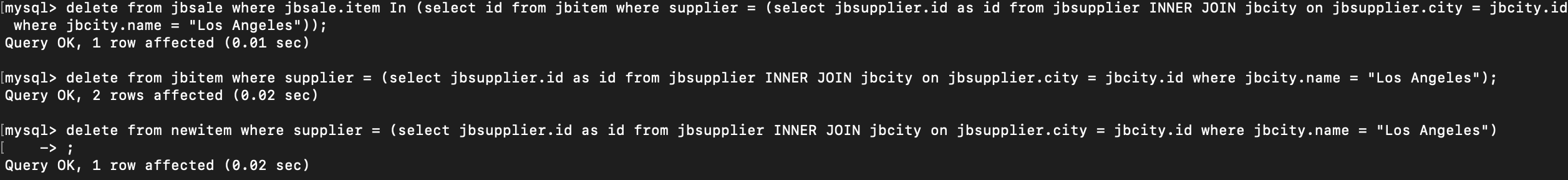




**Motivate why you use the join you do (left, right or inner), and why this is the correct one (unlike the others).**

We have used “INNER JOIN” because INNER JOINS treats the filter condition on equal basis regardless of the position of table while if we would have used LEFT or RIGHT OUTER joins, we had to take care of the position of tables given in join condition. Though with respect to performance there is not much difference between join and where condition except for the fact that for complex queries joins are better to use moreover joins are also better to use in case there are NULL values in the filtered columns

## 19-a)Remove all suppliers in Los Angeles from the table jbsupplier. This will not work right away (you will receive error code 23000) which you will have to solve by deleting some other related tuples. However, do not delete more tuples from other tables than necessary and do not change the structure of the tables, i.e. do not remove foreign keys. Also, remember that you are only allowed to use “Los Angeles” as a constant in your queries, not “199” or “900”.





## 19-b)Explain what you did and why.

Due to foreign key constraints , we have delete the corresponding rows first. Than we find the Los Angeles city from **jbcity** and join it with **jbsupplier** table to find the supplier id and then delete the corresponding rows with supplier id from table **jbsale,jbitem** and **newitem** table and after that we deleted all supplier who supply in Los Angeles from **jbsupplier** table.

## 20)An employee has tried to find out which suppliers that have delivered items that have been sold. He has created a view and a query that shows the number of items sold from a supplier.The employee would also like include the suppliers which has delivered some items, although for whom no items have been sold so far. In other words he wants to list all suppliers, which has supplied any item, as well as the number of these items that have been sold. Help him! Drop and redefine jbsale\_supply to consider suppliers that have delivered items that have never been sold as well.

**Hint:** The above definition of jbsale\_supply uses an (implicit) inner join that removes suppliers that have not had any of their delivered items sold.

