**Solving transportation problem**

Name of student

Name of professor

University

Course

Date

**Part 1 Database design**

1. **Question One**

In this particular section, we are going to take the two table categories that include:

* Orders and sales
* Orders and sales items

**Dataset description:**

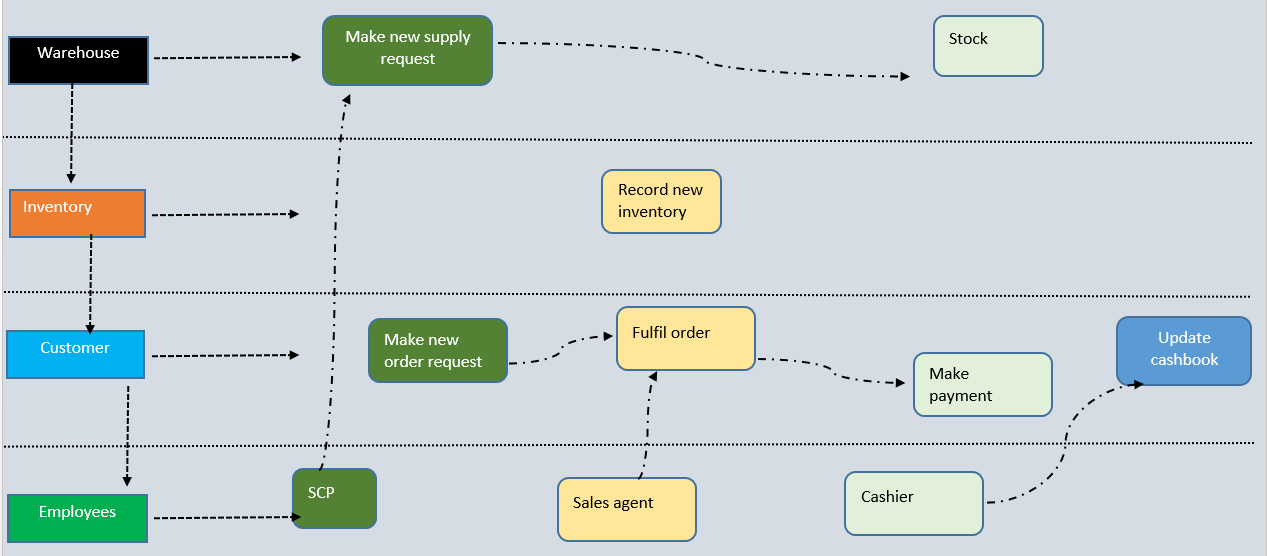
Whenever a client makes an order, the order is captured together with the other order details such as the staff who served the client and the discovered particular product that was served to the client. Other factors taken into consideration are also the product name and the product\_id that was sold.

When building applications, there has to be an establishment within table entities, by what may also be termed as relationships. The above two table classes can be related as follows:

|  |  |
| --- | --- |
| Orders table | Sales items table |

|  |  |  |  |
| --- | --- | --- | --- |
| Order id | Item id | Item id | Order id |
| PRIMARY KEY (PK) | FOREIGN KEY(FK) | PRIMARY KEY(PK) | FOREIGN KEY(FK) |

Whereas the order\_id is the primary key in the orders table, it is the foreign key in the sales table and vice versa. The business process model notation for this activity as shown below:



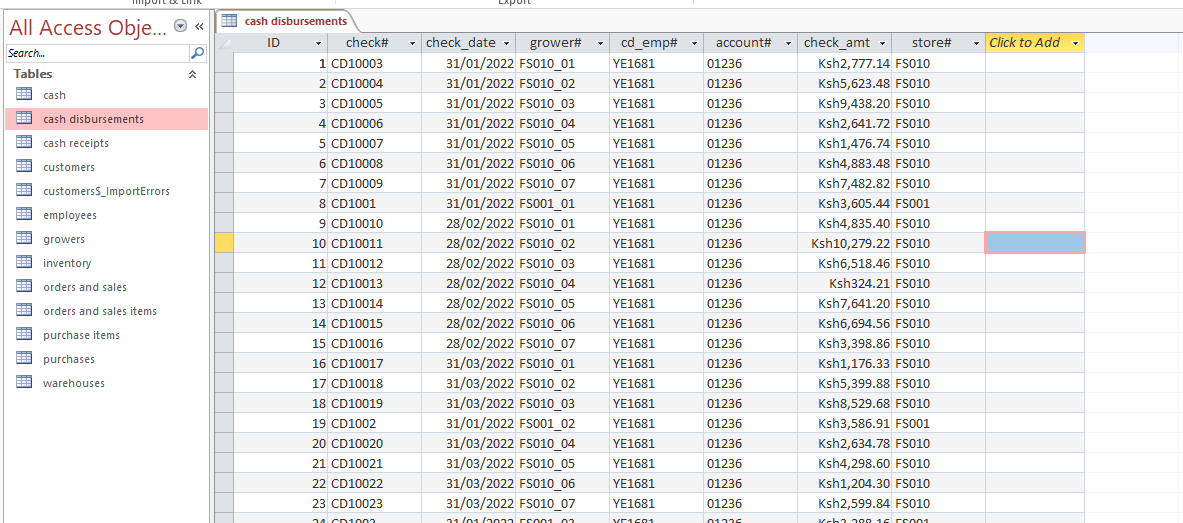
**Question two**

Creating table with business rules:

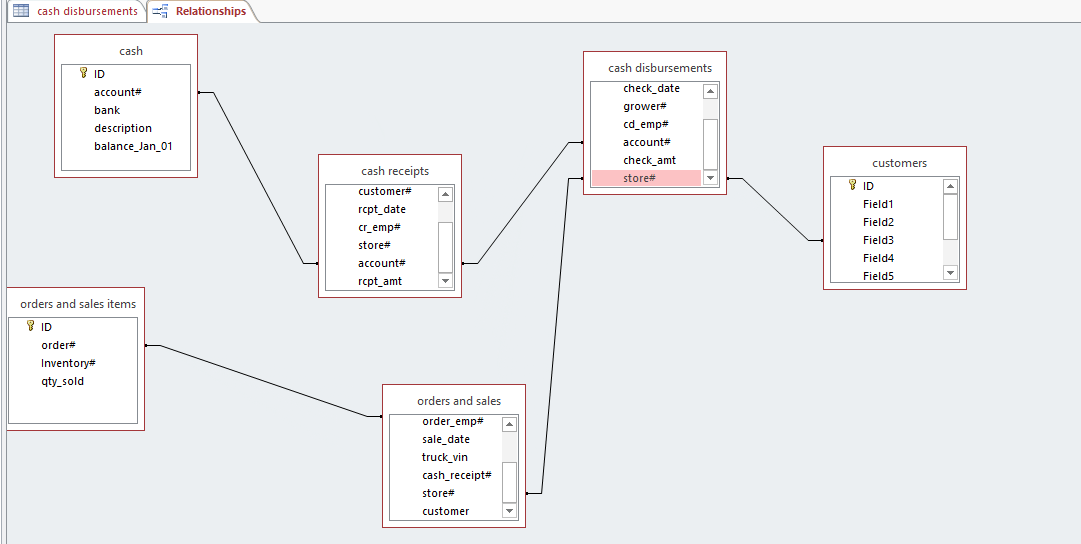
|  |  |  |
| --- | --- | --- |
| Process Steps | Intention | Rules relating to authority, action, or access |
| 1. Acquire | -Make a supply request for product sourcing | -Only SCM staff can make order  -Order is made weekly  -R-order level must be below current 50% |
| 1. Stock | -Record all new supplies in the inventory books | -Only store staff can record inventories  -All inventories must be second proofed |
| 1. Order | -Clients make new orders | -A client can make new orders  -A client can make several orders in a day  -A client can pay for several orders  - Cashiers verifies all payments made |
| 1. Record | -Take all financial records of the payments | -All payments must be recorded  -All sales must be recorded  -Re-curing sales must be recorded  -Daily sales must be approved |

**Question three**

Loading the dataset on Microsoft Access gives the following:



The derived relationship looks as below:



**Part B**

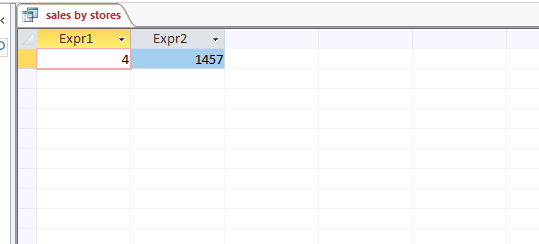
**Database analytics**

Question 4

SELECT store AS Expr1, Count(\*) AS Expr2

FROM [orders and sales]

GROUP BY store;

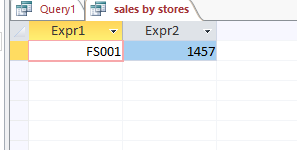


SELECT

store, count(\*)

FROM purchases

group by store;





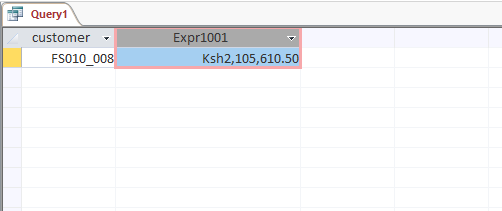
SELECT

customer, sum(rcpt\_amt)

FROM [cash receipts]

group by customer

order by sum(rcpt\_amt) DESC;



SELECT

a.grower, (b.qty\_purch \* purchase\_price) as total\_purchase

FROM purchases a

LEFT OUTER JOIN  [orders and sales items] b

on a.purchase = b.purchase

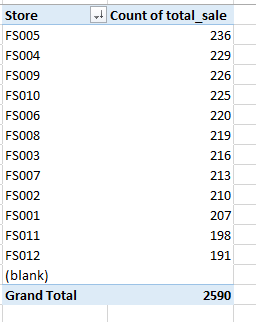
group by total\_purchase, a.grower

order by total\_purchase desc;

SELECT \* FROM PURCHASES\_ITEMS WHERE purchase\_price IS NULL ;

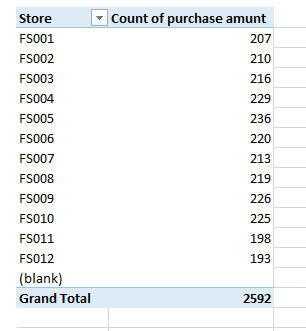
**Question 5**

Sales by store

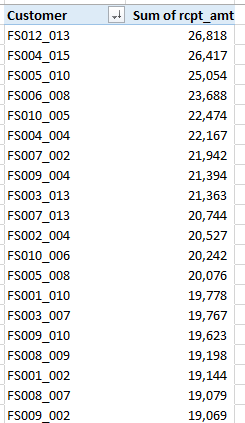


Purchase by store





Sales by customer, arranged from highest to lowest sales

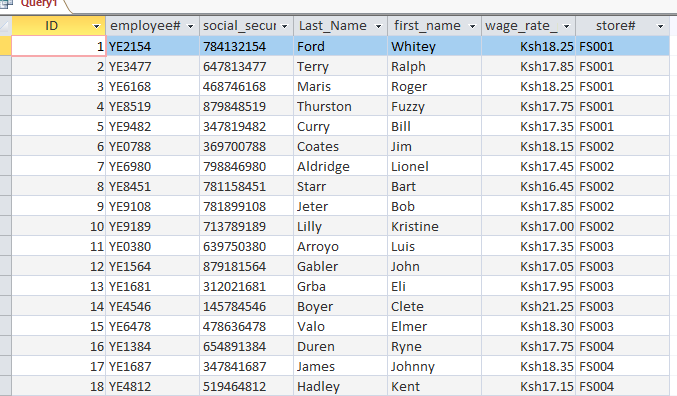


Question 6

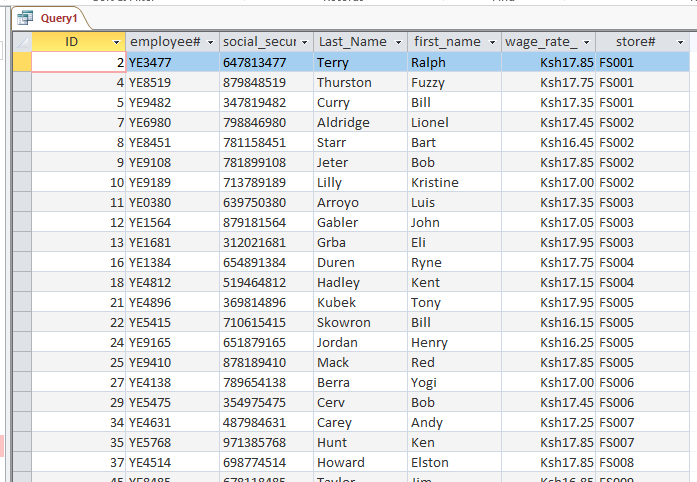
1. Select all employees

SELECT

FROM employees;



6b.) Select all employees where wage rate is below 18 dollars



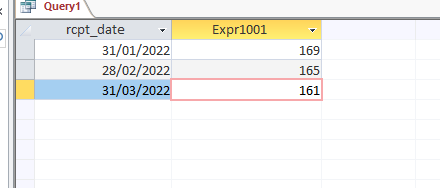
**Question 7**

1. Select all from cash receipt and group by date

SELECT

rcpt\_date, count(\*)

FROM [cash receipts] group by rcpt\_date



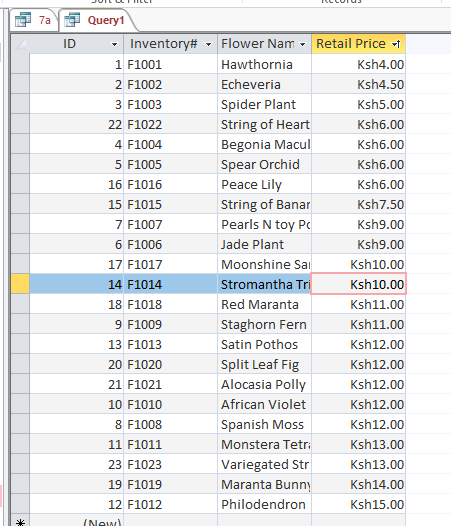
1. Order inventories by inventory descending prices

SELECT

\*

FROM inventory

ORDER BY Retail\_Price ASC;

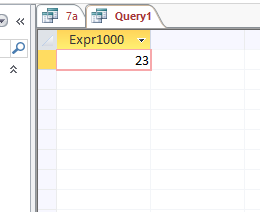


7c. Nested Count query with SQL

SELECT COUNT(\*) FROM

(SELECT COUNT(\*) FROM FROM inventory

WHERE Retail\_Price = 6.0) C



7d. Select all purchases and purchase items

SELECT

FROM purchases INNER JOIN [purchase items] ON purchases.[purchase#] = [purchase items].[purchase#]