

SQL PROJECT

FastKart Retail Store



Created by:



Introduction

You are hired by a chain of online retail stores “Fastkart”. They have provided you with “Fastkart” database and seek answers to the following queries as the results from these queries will help the company in making data-driven decisions that will impact the overall growth of the online retail stores.



EER DIAGRAM OF FASTKART

MySQL WorkBench

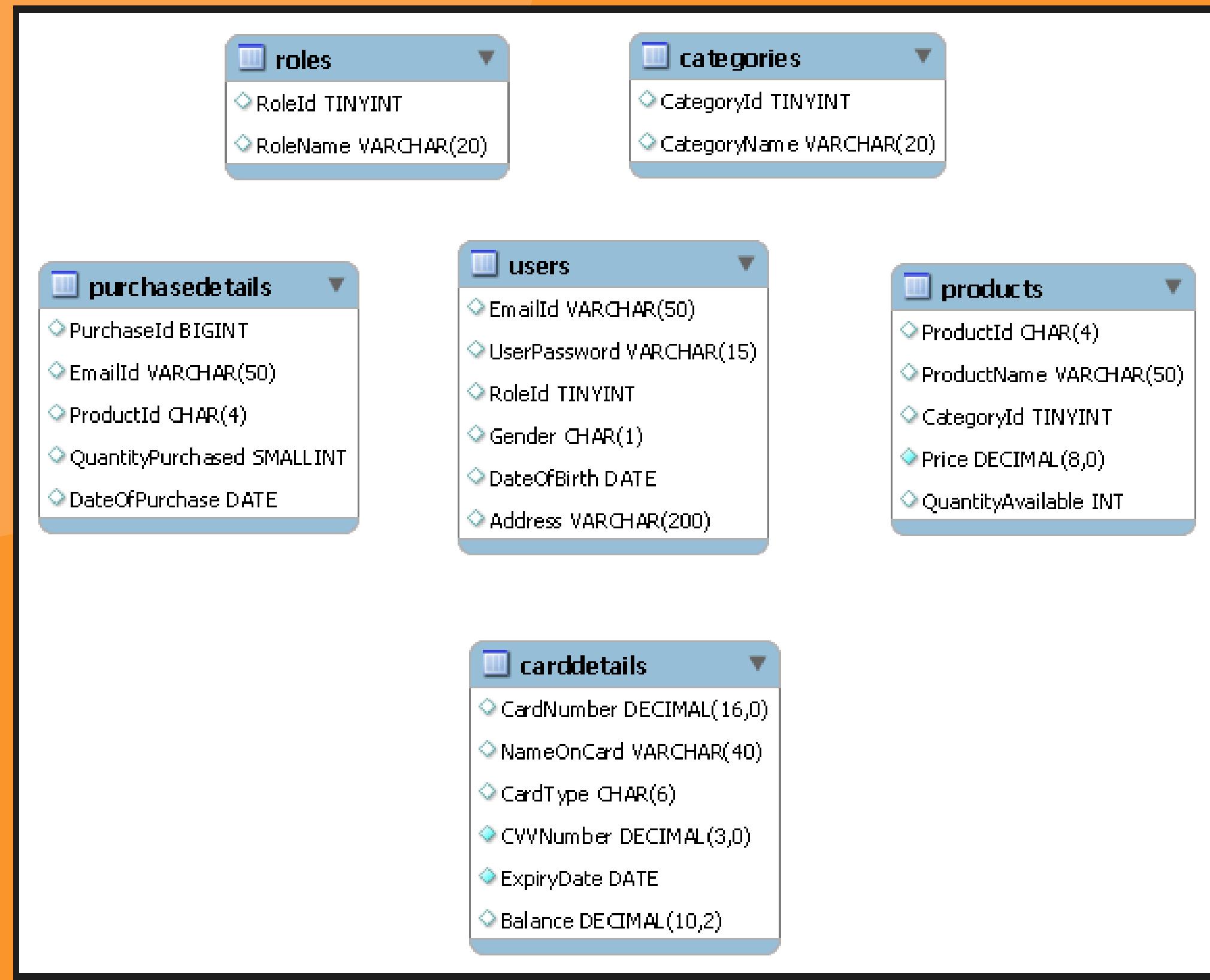


TABLE RELATIONSHIP

There are 6 Tables found in FastKart SQL script where glimpse of some details being executed for each tables: carddetails, categories, products, purchasedetails, roles and users.

CARD DETAILS

	CardNumber	NameOnCard	CardType	CVVNumber	ExpiryDate	Balance
▶	1201611246771470	Cruz	V	171	2026-12-25	13645.00
	1224120265211560	Pirkko	M	771	2027-01-11	14620.00
	1221664512112100	Helen	M	402	2021-06-21	16132.00
	1245674110616670	Mary	M	121	2011-01-04	14071.00
	1251175712010020	Annette	M	606	2022-10-24	15111.00
	1211352607461300	Saveley	V	161	2023-01-05	14120.00
	1307313341777150	Anne	M	614	2011-01-21	16611.00
	1307114461363110	Philip	M	663	2021-01-11	1663.00
	1323151003776600	Parente	V	517	2021-07-22	7532.00

carddetails 8 ×

CATEGORIES

	CategoryId	CategoryName
▶	1	Motors
	2	Fashion
	3	Electronics
	4	Arts
	5	Home
	6	Sporting Goods
	7	Toys

categories 9 ×

PRODUCTS

	ProductId	ProductName	CategoryId	Price	QuantityAvailable
▶	P101	Lamborghini Gallardo Spyder	1	18000000	10
	P102	BMW X1	1	3390000	10
	P103	BMW Z4	1	6890000	10
	P104	Harley Davidson Iron 883	1	700000	10
	P105	Ducati Multistrada	1	2256000	10
	P106	Honda CBR 250R	1	193000	100
	P107	Kenneth Cole Black & White Leather Reversible ...	2	2500	50
	P108	Classic Brooks Brothers 346 Wool Black Sport Coat	2	3079	10
	P109	Ben Sherman Mens Necktie Silk Tie	2	1847	20

products 10 ×

TABLE RELATIONSHIP

There are 6 Tables found in FastKart SQL script where glimpse of some details being executed for each tables: carddetails, categories, products, purchasedetails, roles and users.

PURCHASE DETAILS

	PurchaseId	EmailId	ProductId	QuantityPurchased	DateOfPurchase
▶	1001	Franken@gmail.com	P101	2	2014-01-12
	1002	Franken@gmail.com	P143	1	2014-01-13
	1003	Franken@gmail.com	P112	3	2014-01-14
	1004	Franken@gmail.com	P148	2	2014-01-15
	1005	Franken@gmail.com	P150	1	2014-01-16
	1006	Franken@gmail.com	P134	3	2014-01-16
	1007	SamRocks@gmail.com	P120	4	2013-11-17
	1008	SamRocks@gmail.com	P110	4	2013-11-19
	1009	SamRocks@gmail.com	P112	3	2013-11-20

purchasedetails 11 ×

ROLES

	RoleId	RoleName
▶	1	Admin
	2	Customer

roles 12 ×

USERS

	EmailId	UserPassword	RoleId	Gender	DateOfBirth	Address
▶	Franken@gmail.com	BSBEV@1234	2	F	1976-08-26	Fauntleroy Circus
	Henriot@gmail.com	CACTU@1234	2	F	1971-09-04	Cerrito 333
	Hernadez@gmail.com	CHOPS@1234	2	M	1981-09-18	Hauptstr. 29
	Jablonski@gmail.com	COMMI@1234	2	M	1989-07-21	Av. dos Lusíadas, 23
	Josephs@gmail.com	CONSH@1234	2	F	1963-11-09	Berkeley Gardens 12 Brewery
	Anzio_Don@gmail.com	don@123	1	M	1991-02-24	Surya Bakery, Mysore;Surya Bakery, Mysore-5...
	Karttunen@gmail.com	DRACD@1234	2	M	1963-06-27	Walserweg 21
	Koskitalo@gmail.com	DUMON@1234	2	F	1966-01-28	67, rue des Cinquante Otages
	Labrune@gmail.com	EASTC@1234	2	F	1980-02-09	35 King George

users 13 ×

Q1. List Top 3 products based on QuantityAvailable. (productid, productname, QuantityAvailable). (3 Rows) [Note: Products]

QUERIES

```
select Productid, ProductName, QuantityAvailable from products;
```

```
select Productid, ProductName, QuantityAvailable from products order by QuantityAvailable DESC limit 3;
```

RESULT:

	Productid	ProductName	QuantityAvailable
▶	P128	Wooden photo frame	200
	P143	Adidas Shoes	150
	P144	Tennis racket	150



Q2. Display EmailId of those customers who have done more than ten purchases. (EmailId, Total_Transactions). (5 Rows) [Note: Purchasedetails, products]



QUERIES

```
select pd.EmailId, pd.QuantityPurchased as Total_Transactions  
from PurchaseDetails pd  
inner join Products p  
on pd.ProductId = p.ProductId  
group by EmailId  
HAVING count(QuantityPurchased) > 10  
order by Total_Transactions DESC;
```

RESULT:

	EmailId	Total_Transactions
▶	Timothy@gmail.com	4
	Franken@gmail.com	2
	Matti@gmail.com	2
	Roland@gmail.com	1
	Helvetica@gmail.com	1

Q3. List the Total QuantityAvailable category wise in descending order. (Name of the category, QuantityAvailable) (7 Rows) [Note: products, categories]

QUERIES

```
Select c.CategoryName as NameOfTheCategory, p.QuantityAvailable as TotalQuantityAvailable  
from Products p  
INNER JOIN Categories c  
ON p.CategoryId = c.CategoryId  
group by c.CategoryName  
Order by p.QuantityAvailable DESC;
```

RESULT:

	NameOfTheCategory	TotalQuantityAvailable
▶	Arts	100
▶	Home	100
▶	Sporting Goods	100
▶	Toys	100
▶	Electronics	70
▶	Fashion	50
▶	Motors	10

Q4. Display ProductId, ProductName, CategoryName, Total_Purchased_Quantity for the product which has been sold maximum in terms of quantity? (1 Row)
[Note: purchasedetails, products, categories]

QUERIES

```
SELECT p.ProductId,p.ProductName,c.CategoryName, sum(pd.QuantityPurchased) as Total_Purchased_Quantity  
FROM Categories c  
INNER JOIN Products p  
ON c.CategoryId=p.CategoryId  
INNER JOIN PurchaseDetails pd  
ON p.ProductId=pd.ProductId;
```

RESULT:

ProductId	ProductName	CategoryName	Total_Purchased_Quantity
P101	Lamborghini Gallardo Spyder	Motors	344



Q5. Display the number of male and female customers in fastkart. (2 Rows)

[Note: roles, users]

QUERIES

```
select count(*) as NoOfMalesOrFemales  
From Users  
Group by Gender;
```

RESULT:

Result Grid | Filter Rows: Export: Wrap Cell Content:

	NoOfMalesOrFemales
▶	40
	45

Q6. Display ProductId, ProductName, Price and Item_Classes of all the products where Item_Classes are as follows: If the price of an item is less than 2,000 then “Affordable”, If the price of an item is in between 2,000 and 50,000 then “High End Stuff”, If the price of an item is more than 50,000 then “Luxury”. (57 Rows)



QUERIES

```
SELECT ProductId,ProductName,Price,  
CASE  
WHEN Price < 2000 THEN 'Affordable'  
WHEN Price BETWEEN 2000 AND 50000 THEN 'High And Stuff'  
WHEN Price > 50000 THEN 'Luxury'  
ELSE Price  
END as ItemClasses  
FROM Products  
ORDER BY Price desc;
```

RESULT:

	ProductId	ProductName	Price	ItemClasses
▶	P101	Lamborghini Gallardo Spyder	18000000	Luxury
	P103	BMW Z4	6890000	Luxury
	P102	BMW X1	3390000	Luxury
	P105	Ducati Multistrada	2256000	Luxury
	P104	Harley Davidson Iron 883	700000	Luxury
	P106	Honda CBR 250R	193000	Luxury
	P141	Turner Sultan 29er Large	147613	Luxury
	P120	Apple MAcbook Pro	56800	Luxury
	P115	Apple IPhone 5s 16GB	52750	Luxury

Q7. Write a query to display ProductId, ProductName, CategoryName, Old_Price(price) and New_Price as per the following criteria a. If the category is “Motors”, decrease the price by 3000 b. If the category is “Electronics”, increase the price by 50 c. If the category is “Fashion”, increase the price by 150 For the rest of the categories price remains same. Hint: Use case statement, there should be no permanent change done in table/DB. (57 Rows) [Note: products, categories]

QUERIES

```
SELECT p.ProductId,p.ProductName,c.CategoryName,p.Price,  
CASE CategoryName  
WHEN 'Motors' THEN Price - 3000  
WHEN 'Electronics' THEN Price + 50  
WHEN 'Fashion' THEN Price + 150  
ELSE Price  
END as NewPrice  
FROM Products p  
INNER JOIN Categories c  
ON p.CategoryId=c.CategoryId  
ORDER BY ProductId;
```

RESULT:

	ProductId	ProductName	CategoryName	Price	NewPrice
▶	P101	Lamborghini Gallardo Spyder	Motors	18000000	17997000
	P102	BMW X1	Motors	3390000	3387000
	P103	BMW Z4	Motors	6890000	6887000
	P104	Harley Davidson Iron 883	Motors	700000	697000
	P105	Ducati Multistrada	Motors	2256000	2253000
	P106	Honda CBR 250R	Motors	193000	190000
	P107	Kenneth Cole Black & White Leather Reversible ...	Fashion	2500	2650
	P108	Classic Brooks Brothers 346 Wool Black Sport Coat	Fashion	3079	3229
	P109	Ben Sherman Mens Necktie Silk Tie	Fashion	1847	1997



Q8. Display the percentage of females present among all Users.
(Round up to 2 decimal places) Add “%” sign while displaying the
percentage. (1 Row) [Note: users]

QUERIES

```
SELECT ROUND(count(Gender='F')/(select count(Gender) from Users)*100, 2)  
as Female ,'%'  
From Users  
Where Gender ='F';
```

RESULT:

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	Female	%			
▶	47.06	%			



Q9. Display the average balance for both card types for those records only where CVVNumber > 333 and NameOnCard ends with the alphabet “e”. (2 Rows) [Note: carddetails]

QUERIES

```
SELECT CardType,  
avg (Balance)  
FROM CardDetails  
WHERE CVVNumber >333  
AND NameOnCard like '%e'  
GROUP by CardType;
```

RESULT:

	CardType	avg (Balance)
▶	M	11740.000000
V		7028.250000

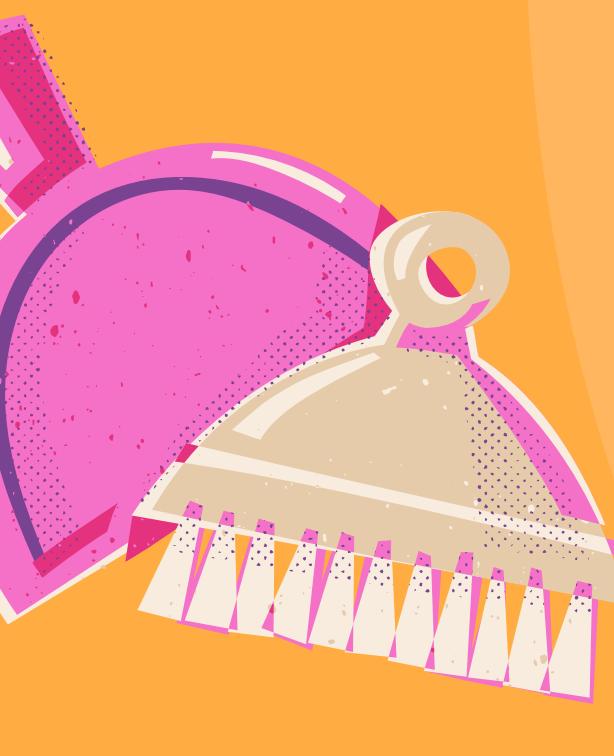


Q10. What is the 2nd most valuable item available which does not belong to the "Motor" category. Value of an item = Price * QuantityAvailable. Display ProductName, CategoryName, value. (1 Row) [Note: products, categories]

QUERIES

```
SELECT p.ProductName, c.CategoryName, sum(p.Price*p.QuantityAvailable) as VALUE  
From Products p  
INNER JOIN Categories c  
On p.CategoryId=c.CategoryId  
WHERE CategoryName <> 'Motors'  
ORDER by ProductName;
```

RESULT:



	ProductName	CategoryName	VALUE
▶	Kenneth Cole Black & White Leather Reversible ...	Fashion	55181080



SQL FASTKART PRESENTATION

THANK YOU FOR
YOUR TIME

Created by:

RISHI KALPA MUKHERJEE