

```
--1
/*Create a stored procedure to show each vendor's products, separated by commas. (u can
use
STRING_AGG and TRIM functions)*/
--used functions above to show vendors products
create procedure vendor_product as
begin
select v.vend_id, v.vend_name, v.vend_country, p.prod_name from Vendors v full join
(select vend_id, STRING_AGG(TRIM(prod_name), ',') as prod_name from Products group by
vend_id) p on v.vend_id = p.vend_id
end
exec vendor_product
drop procedure vendor_product
```

	vend_id	vend_name	vend_country	prod_name
1	BRE02	Bear Emporium	USA	NULL
2	BRS01	Bears R Us	USA	8 inch teddy bear,12 inch teddy bear,18 inch teddy b...
3	DLL01	Doll House Inc.	USA	Raggedy Ann,Fish bean bag toy,Bird bean bag toy,....
4	FNG01	Fun and Games	England	King doll,Queen doll
5	FRB01	Furball Inc.	USA	NULL
6	JTS01	Jouets et ours	France	NULL

```
--2
/*Create a procedure that updates emails for null values in Customers table. It has to be
started
from 'kztoys', @nameofcustomer without spaces. (u can use CONCAT, LOWER and REPLACE
functions)*/
--updating data in null columns
go
create procedure email_update as
begin
update Customers set cust_email=CONCAT('kztoys','@',LOWER(REPLACE(cust_name, '
','')),'.com') where cust_email IS NULL
end
exec email_update
drop procedure email_update
```

Messages

(0 rows affected)

Completion time: 2022-04-29T01:25:15.3423790+06:00

```
--3
/*Create a stored procedure that shows the customers and regularities of their
visits to the market. The customer is said to be 'REGULAR CUSTOMER' if he/she bought
at least 2 things, otherwise 'SELDOM CUSTOMER'. Show all the info with this description.
(u can use CASE statement)*/
go
```

```

create procedure description_insert as
begin
    declare @status varchar(50)
    select c.cust_id, c.cust_name, c.cust_address, c.cust_email, o.counts, (case when
o.counts > 1 then 'regular customer' else 'seldom customer' end) as 'description' from
Customers c, (select count(*) as 'counts', cust_id from Orders group by cust_id) o where
c.cust_id=o.cust_id
end

exec description_insert
drop procedure description_insert

```

	cust_id	cust_name	cust_address	cust_email	counts	description
1	1000000001	Village Toys	200 Maple Lane	mailto:sales@villagetoys.com	2	regular customer
2	1000000003	Fun4All	1 Sunny Place	<mailto:jjones@fun4all.com	1	seldom customer
3	1000000004	Fun4All	829 Riverside Drive	<mailto:dstephens@fun4all.com	1	seldom customer
4	1000000005	The Toy Store	4545 53rd Street	kztoys@thetoystore.com	1	seldom customer

```

--4
/*Create a stored procedure that shows all the id, name, country as well as description
of all the vendors.
The description must indicate whether or not the length of the country Odd or Even.*/
-- new column depending on len of vend_name
go
create procedure vendor_description_insert as
begin
    select v.vend_id, v.vend_name, v.vend_country, (case when len(vend_country)%2=0
then 'even' else 'odd' end) as vend_description from Vendors v
end
exec vendor_description_insert
drop procedure vendor_description_insert

```

	vend_id	vend_name	vend_country	vend_description
1	BRE02	Bear Emporium	USA	odd
2	BRS01	Bears R Us	USA	odd
3	DLL01	Doll House Inc.	USA	odd
4	FNG01	Fun and Games	England	odd
5	FRB01	Furball Inc.	USA	odd
6	JTS01	Jouets et ours	France	even

```

--5
/*Create a stored procedure to display full cost for product by input id.*/
go
create procedure full_cost @prod_id varchar(50), @order_num int as
begin
declare @output varchar(50)
set @output = concat('Full cost is ', (SELECT CONVERT(varchar, item_price*quantity) from
OrderItems where prod_id=@prod_id AND order_num=@order_num))
print @output
end

```

```
exec full_cost @prod_id='BR01', @order_num=20006;  
drop procedure full_cost
```

Messages

Full cost is 119.80

Completion time: 2022-04-29T01:27:13.4802393+06:00

```
--6  
/*Create a stored procedure that gives a message  
'There are some vendors with unknown locations'  
if exists unknown locations in vendors table. Otherwise it should show 'Information about  
location is filled'.*/  
-- checking is there nulls in vend city  
go  
create procedure messagee as  
begin  
declare @output varchar(50)  
set @output = (case when (select count(*) from Vendors where vend_city is null) > 0 then  
'There are some vendors with unknown locations' else 'Information about location is  
filled' end)  
print @output  
end  
exec messagee  
drop procedure messagee
```

Messages

Information about location is filled

Completion time: 2022-04-29T01:27:40.8076443+06:00

```
--7  
/*Create a stored procedure that shows is there vendor in the state where customer  
lives.*/  
-- checking are hte from same state?  
go  
create procedure about_city as  
begin  
/*create view tmp as  
select c.cust_name, c.cust_state, v.vend_state from  
Customers c, Orders o, OrderItems oi, Products p, Vendors v  
where c.cust_id = o.cust_id and o.order_num = oi.order_num and oi.prod_id = p.prod_id and  
p.vend_id = v.vend_id*/  
select distinct(c.cust_name), (case when c.cust_state = v.vend_state then 'YES' else 'NO'  
end) as vend from Customers c, Vendors v  
end  
exec about_city  
drop procedure about_city
```

Results Messages		
	cust_name	vend
1	Fun4All	NO
2	Kids Place	NO
3	Kids Place	YES
4	The Toy Store	NO
5	Village Toys	NO
6	Village Toys	YES

```
--8
/*Create a stored procedure that shows all the products whose price is located between
average price and second higher average price of the products.
The second higher average price is located exactly in the middle of the mean value and
max value price of the products.*/
go
create procedure avg_price as
begin
declare @avg1 int
set @avg1 = (select avg(prod_price) from Products)
declare @max int
set @max = (select max(prod_price) from Products)
declare @avg2 int
set @avg2 = (select avg(prod_price) from Products where prod_price between @avg1 and
@max)
select * from Products where prod_price between @avg1 and @avg2
end

exec avg_price
drop procedure avg_price
```

Results Messages						
	prod_id	vend_id	prod_name	amount	prod_price	prod_desc
1	BR02	BRS01	12 inch teddy bear	18	8.99	12 inch teddy bear, comes with cap and jacket

```
--9
/*Create a stored procedure that shows customer ids, number of items that they bought,
total amount of money that they paid, and to one column write if it is max amount of
money - 'maximum',
if second maximum write 'second maximum' and otherwise write 'usual'.*/
-- finding all boughts of each customer (here i couldnt find max and second max) (it
works if take off last column which should find is it max or not)
go
create procedure customers_data as
begin
declare @max int
declare @s_max int
set @max = (select max(MON) from (select sum(oi.item_price*oi.quantity) as MON from
Customers c, Orders o, OrderItems oi where c.cust_id=o.cust_id and
o.order_num=oi.order_num group by c.cust_id) a)
```

```

set @s_max = (select max(MON) from (select sum(oi.item_price*oi.quantity) as MON from
Customers c, Orders o, OrderItems oi where c.cust_id=o.cust_id and
o.order_num=oi.order_num group by c.cust_id) a where MON != @max)
select c.cust_id, count(*) as num_items, sum(oi.item_price*oi.quantity) as MON, case when
MON = @max then 'maximum' when MON = @s_max then 'second_max' else 'usual' end as status
from Customers c, Orders o, OrderItems oi where c.cust_id=o.cust_id and
o.order_num=oi.order_num group by c.cust_id
end

```

exec customers_data

	cust_id	num_items	MON	status
1	1000000001	5	3515.50	usual
2	1000000003	3	329.60	usual
3	1000000004	5	1696.00	usual
4	1000000005	5	189.60	usual

--10

-- here i am finding all products and prderitems of defined vendor

```

go
create procedure vendors_products as
begin
select * from
OrderItems oi, Products p, Vendors v
where oi.prod_id = p.prod_id and p.vend_id = v.vend_id and v.vend_id = 'DLL01'
end

```

exec vendors_products

	order_num	order_item	prod_id	quantity	item_price	prod_id	vend_id	prod_name	amount	prod_price	prod_desc	vend_id	vend_name	vend_address	vend_city	vend_state	vend_zip	vend
1	20007	2	BNBG01	100	2.99	BNBG01	DLL01	Fish bean bag toy	50	3.49	Fish bean bag toy, complete with bean bag worms ...	DLL01	Doll House Inc.	555 High Street	Dollsville	CA	99999	USA
2	20007	3	BNBG02	100	2.99	BNBG02	DLL01	Bird bean bag toy	200	3.49	Bird bean bag toy, eggs are not included	DLL01	Doll House Inc.	555 High Street	Dollsville	CA	99999	USA
3	20007	4	BNBG03	100	2.99	BNBG03	DLL01	Rabbit bean bag toy	140	3.49	Rabbit bean bag toy, comes with bean bag carrots	DLL01	Doll House Inc.	555 High Street	Dollsville	CA	99999	USA
4	20007	5	RGAN01	50	4.49	RGAN01	DLL01	Raggedy Ann	45	4.99	18 inch Raggedy Ann doll	DLL01	Doll House Inc.	555 High Street	Dollsville	CA	99999	USA
5	20008	1	BNBG01	5	4.99	BNBG01	DLL01	Fish bean bag toy	50	3.49	Fish bean bag toy, complete with bean bag worms ...	DLL01	Doll House Inc.	555 High Street	Dollsville	CA	99999	USA
6	20008	3	BNBG02	10	3.49	BNBG02	DLL01	Bird bean bag toy	200	3.49	Bird bean bag toy, eggs are not included	DLL01	Doll House Inc.	555 High Street	Dollsville	CA	99999	USA
7	20008	4	BNBG03	10	3.49	BNBG03	DLL01	Rabbit bean bag toy	140	3.49	Rabbit bean bag toy, comes with bean bag carrots	DLL01	Doll House Inc.	555 High Street	Dollsville	CA	99999	USA
8	20009	1	BNBG01	250	2.49	BNBG01	DLL01	Fish bean bag toy	50	3.49	Fish bean bag toy, complete with bean bag worms ...	DLL01	Doll House Inc.	555 High Street	Dollsville	CA	99999	USA
9	20009	2	BNBG02	250	2.49	BNBG02	DLL01	Bird bean bag toy	200	3.49	Bird bean bag toy, eggs are not included	DLL01	Doll House Inc.	555 High Street	Dollsville	CA	99999	USA
10	20009	3	BNBG03	250	2.49	BNBG03	DLL01	Rabbit bean bag toy	140	3.49	Rabbit bean bag toy, comes with bean bag carrots	DLL01	Doll House Inc.	555 High Street	Dollsville	CA	99999	USA