

Alimgozha Adil

Lab Procedures

```
--1.Create a stored procedure to show each vendor's products, separated by commas.(u can use STRING_AGGand TRIM functions)
CREATE PROCEDURE VENDOR_PRODUCT
AS
BEGIN
    SELECT V.VEND_ID, VEND_NAME, VEND_COUNTRY, PRODUCTS
    FROM Vendors AS V
    LEFT JOIN (
        SELECT VEND_ID, STRING_AGG (CONVERT(NVARCHAR(max), TRIM(PROD_NAME)), ', ') AS PRODUCTS
        FROM Products
        GROUP BY vend_id) AS P
    ON V.VEND_ID = P.vend_id
END

EXEC VENDOR_PRODUCT
```

100 %

Результаты

	VEND_ID	VEND_NAME	VEND_COUNTRY	PRODUCTS
1	BRE02	Bear Emporium	USA	NULL
2	BRS01	Bears R Us	USA	8 inch teddy bear, 12 inch teddy bear, 18 inch teddy bear
3	DLL01	Doll House Inc.	USA	Fish bean bag toy, Bird bean bag toy, Rabbit bean bag toy, Raggedy Ann
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)*/
CREATE PROCEDURE UPDATE_EMAIL
@nameofcustomer VARCHAR(50)

AS
BEGIN
    UPDATE CUSTOMERS
    SET cust_email = CONCAT('kztoys.', REPLACE(LOWER(@nameofcustomer), ' ', ''), '.com')
    WHERE cust_name = @nameofcustomer
END

EXEC UPDATE_EMAIL @nameofcustomer = 'Kids Place'
EXEC UPDATE_EMAIL @nameofcustomer = 'The Toy Store'

SELECT * FROM Customers
```

100 %

Результаты

	cust_id	cust_name	cust_address	cust_city	cust_state	cust_zip	cust_country	cust_contact	cust_email
1	1000000001	Village Toys	200 Maple Lane	Detroit	MI	44444	USA	John Smith	mailto:sales@villagetoy.com
2	1000000002	Kids Place	333 South Lake Drive	Columbus	OH	43333	USA	Michelle Green	kztoys.@kidsplace.com
3	1000000003	Fun4All	1 Sunny Place	Muncie	IN	42222	USA	Jim Jones	<mailto:jjones@fun4all.com
4	1000000004	Fun4All	829 Riverside Drive	Phoenix	AZ	88888	USA	Denise L. Stephens	<mailto:dstephens@fun4all.com
5	1000000005	The Toy Store	4545 53rd Street	Chicago	IL	54545	USA	Kim Howard	kztoys.@thetoystore.com

```

/*3.Create a stored procedure that showsthe 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 CASEstatement)*/

```

```

CREATE PROCEDURE REGULAR_SELDOM_CUST
AS
BEGIN
    SELECT C.cust_id, cust_name, cust_address, cust_city, cust_email, O.AMOUNT_OF_ORDERS,
    CASE
        WHEN O.AMOUNT_OF_ORDERS = 1
        THEN 'seldom customer'
        ELSE 'regular customer'
    END AS CUSTOMER_DESCRIPTION
    FROM Customers C
    JOIN (
        SELECT cust_id, COUNT(order_num) AS AMOUNT_OF_ORDERS
        FROM ORDERS
        GROUP BY cust_id
    ) AS O
    ON C.cust_id = O.cust_id
END

```

```
EXEC REGULAR_SELDOM_CUST
```

100 %

Результаты

Сообщения

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

```

/*4.Create a stored procedure that shows all the id, name, countryas well as description of all the vendors.
The description must indicate whether or not the length of the countryOddor Even.*/

```

```

CREATE PROCEDURE EVEN_ODD
AS
BEGIN
    SELECT vend_id, vend_name, vend_country,
    CASE
        WHEN LEN(vend_country) % 2 = 0
        THEN 'even'
        ELSE 'odd'
    END AS VEND_COUNTRY_DESCRIPTION
    FROM Vendors
END

```

```
EXEC EVEN_ODD
```

100 %

Результаты

Сообщения

	vend_id	vend_name	vend_country	VEND_COUNTRY_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. */
CREATE PROCEDURE PRICE
    @PROD_ID CHAR(10),
    @ORDER_NUM INT
AS
BEGIN
    DECLARE @FULL_COST FLOAT = (SELECT item_price * quantity
    FROM OrderItems
    WHERE PROD_ID = @PROD_ID AND ORDER_NUM = @ORDER_NUM)

    PRINT 'FULL COST ' + CAST(@FULL_COST AS VARCHAR(50))
END

EXEC PRICE @PROD_ID = 'BR01', @ORDER_NUM = 20006
```

100 %

Сообщения

FULL COST 119.8

```
/*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'.*/
CREATE PROCEDURE [MESSAGE]
AS
BEGIN
    DECLARE @COUNT_NULL_LOC INT = (SELECT COUNT(*) FROM(
    SELECT vend_address, vend_city, vend_state, vend_country
    FROM VENDORS
    WHERE vend_address IS NULL OR vend_city IS NULL OR vend_state IS NULL OR vend_country IS NULL) AS COUNT_NULL_LOC_T)
    IF @COUNT_NULL_LOC = 0
        PRINT 'Information about location is filled'
    ELSE
        PRINT 'There are some vendors with unknown locations'
END

EXEC [MESSAGE]
```

100 %

Сообщения

There are some vendors with unknown locations

```

/*7.Create a stored procedure that shows is there vendor in the state where customer lives.*/
CREATE PROCEDURE SAME_STATE
AS
BEGIN
    SELECT cust_name,
    CASE
        WHEN cust_state IN (
            SELECT vend_state
            FROM Vendors)
        THEN 'YES'
        ELSE 'NO'
    END AS VEND
    FROM Customers
END

EXEC SAME_STATE

```

91 %

Результаты Сообщения

	cust_name	VEND
1	Village Toys	YES
2	Kids Place	YES
3	Fun4All	NO
4	Fun4All	NO
5	The Toy Store	NO

```

/*8.Createa stored procedure that shows all the products whoseprice 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.*/
CREATE PROCEDURE PROD
AS
BEGIN
    DECLARE @AVG FLOAT = (SELECT AVG(prod_price) FROM PRODUCTS)
    DECLARE @MAX FLOAT = (SELECT MAX(prod_price) FROM PRODUCTS)
    DECLARE @SEC_HIGH_AVG FLOAT = (
        SELECT AVG(prod_price) FROM Products
        WHERE prod_price BETWEEN @AVG AND @MAX)
    SELECT * FROM PRODUCTS
    WHERE prod_price BETWEEN @AVG AND @SEC_HIGH_AVG
END

EXEC PROD

```

100 %

Результаты Сообщения

	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
2	RYL01	FNG01	King doll	120	9.49	12 inch king doll with royal garments and crown
3	RYL02	FNG01	Queen doll	18	9.49	12 inch queen doll with royal garments and crown

```

/*9.Create a storedprocedure thatshows customer ids, number of items that they bought,
total amount of moneythatthey paid, and to one column write if it is max amount of money -'maximum',
if second maximum write 'second maximum' and otherwise write 'usual'.*/
CREATE PROCEDURE CUST_MONEY
AS
BEGIN

    DECLARE @MAX FLOAT = (
        SELECT MAX(MONEY) FROM(
            SELECT SUM(OI.item_price * OI.quantity) AS MONEY
            FROM Customers C
            INNER JOIN Orders O
            ON C.cust_id = O.cust_id
            INNER JOIN OrderItems OI
            ON O.order_num = OI.order_num
            GROUP BY C.cust_id) AS MONEY_T
        )

    DECLARE @SECOND_MAX FLOAT = (
        SELECT MAX(MONEY) FROM(
            SELECT SUM(OI.item_price * OI.quantity) AS MONEY
            FROM Customers C
            INNER JOIN Orders O
            ON C.cust_id = O.cust_id
            INNER JOIN OrderItems OI
            ON O.order_num = OI.order_num
            GROUP BY C.cust_id) AS MONEY_T
        WHERE MONEY NOT IN (SELECT MAX(MONEY) FROM(
            SELECT SUM(OI.item_price * OI.quantity) AS MONEY
            FROM Customers C
            INNER JOIN Orders O
            ON C.cust_id = O.cust_id
            INNER JOIN OrderItems OI

```

```

        ON C.cust_id = O.cust_id
        INNER JOIN OrderItems OI
        ON O.order_num = OI.order_num
        GROUP BY C.cust_id) AS MONEY_T
WHERE MONEY NOT IN (SELECT MAX(MONEY) FROM(
    SELECT SUM(OI.item_price * OI.quantity) AS MONEY
    FROM Customers C
    INNER JOIN Orders O
    ON C.cust_id = O.cust_id
    INNER JOIN OrderItems OI
    ON O.order_num = OI.order_num
    GROUP BY C.cust_id) AS MONEY_T)
)

SELECT C.cust_id, COUNT(O.order_num) AS NUM_ITEMS, SUM(OI.item_price * OI.quantity) AS MONEY,
CASE
    WHEN SUM(OI.item_price * OI.quantity) = @MAX
    THEN 'MAXIMUM'
    WHEN SUM(OI.item_price * OI.quantity) = @SECOND_MAX
    THEN 'SECOND MAXIMUM'
    ELSE 'USUAL'
END AS DER

FROM Customers C
INNER JOIN Orders O
ON C.cust_id = O.cust_id
INNER JOIN OrderItems OI
ON O.order_num = OI.order_num
GROUP BY C.cust_id
END

EXEC CUST_MONEY

```

91 %

Результаты Сообщения

	cust_id	NUM_ITEMS	MONEY	DER
1	1000000001	5	3515.50	MAXIMUM
2	1000000003	3	329.60	USUAL
3	1000000004	5	1696.00	SECOND MAXIMUM
4	1000000005	5	189.60	USUAL

```

/*10.Your procedure.
Create procedure that takes customer's id and shows the customer's name, quantity of all items and total price*/
CREATE PROCEDURE ORDER_INFO
@CUST_ID INT

AS
BEGIN
    SELECT cust_name, SUM(quantity) AS QUANTITY_ALL_ITEMS, SUM((quantity * item_price)) AS TOTAL_PRICE
    FROM Customers C
    JOIN Orders O
    ON C.cust_id = O.cust_id
    JOIN OrderItems OI
    ON O.order_num = OI.order_num
    WHERE C.cust_id = @CUST_ID
    GROUP BY cust_name
END

EXEC ORDER_INFO @CUST_ID = 1000000001

```

91 %

Результаты Сообщения

	cust_name	QUANTITY_ALL_ITEMS	TOTAL_PRICE
1	Village Toys	950	3515.50