CREATE DATABASE cosmos;

SET FOREIGN\_KEY\_CHECKS = 0;

CREATE TABLE Client(

id INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

name VARCHAR(50),

surname VARCHAR(50),

phone VARCHAR(12),

phone2 VARCHAR(12),

adress VARCHAR(200),

email VARCHAR(50)

);

CREATE TABLE Client\_Archive(

id INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

name VARCHAR(50),

surname VARCHAR(50),

phone VARCHAR(50),

adress VARCHAR(200),

email VARCHAR(50)

);

CREATE TABLE Client\_Request(

id INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

client\_id INT,

FOREIGN KEY (client\_id) REFERENCES Client(id),

request VARCHAR(200),

checked BOOLEAN DEFAULT FALSE,

approved BOOLEAN DEFAULT FALSE

);

CREATE TABLE Contract(

id INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

contract VARCHAR(200),

payment BOOLEAN DEFAULT FALSE,

begin\_date DATE,

end\_date DATE

);

CREATE TABLE Contract\_Archive(

id INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

contract VARCHAR(200),

begin\_date DATE,

end\_date DATE

);

CREATE TABLE Client\_Order(

id INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

client\_id INT,

request\_id INT,

contract VARCHAR(200),

begin\_date DATE,

end\_date DATE,

payment BOOLEAN DEFAULT FALSE,

FOREIGN KEY (client\_id) REFERENCES Client(id)

);

CREATE TABLE Product(

id INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

name VARCHAR(200),

category VARCHAR(50) NOT NULL,

price INT,

count INT

);

CREATE TABLE Order\_Product(

id INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

order\_id INT,

product\_id INT,

count INT,

rest INT,

FOREIGN KEY (order\_id) REFERENCES Client\_Order(id),

FOREIGN KEY (product\_id) REFERENCES Product(id)

);

CREATE TABLE Invoice(

id INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

order\_id INT,

date\_create DATE,

FOREIGN KEY (order\_id) REFERENCES Client\_Order(id),

agreed BOOLEAN DEFAULT FALSE

);

CREATE TABLE Invoice\_Product(

id INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

invoice\_id INT,

product\_id INT,

FOREIGN KEY (invoice\_id) REFERENCES Invoice(id),

FOREIGN KEY (product\_id) REFERENCES Product(id),

count INT,

loaded BOOLEAN DEFAULT FALSE

);

CREATE TABLE Request\_Product(

id INT PRIMARY KEY NOT NULL AUTO\_INCREMENT,

request\_id INT,

product\_id INT,

count INT,

FOREIGN KEY (request\_id) REFERENCES Client\_Request(id),

FOREIGN KEY (product\_id) REFERENCES Product(id)

);

CREATE TABLE Users(

login VARCHAR(16) PRIMARY KEY NOT NULL,

password VARCHAR(20),

role VARCHAR(20)

);

CREATE VIEW Statement\_Invoice AS

SELECT Client.adress, Client.name AS client\_name, Client.surname, Client.phone, Product.name AS product\_name, Invoice\_Product.count, Invoice\_Product.loaded

FROM (((Invoice INNER JOIN Client\_Order ON Invoice.order\_id = Client\_Order.id)

INNER JOIN Client ON Client\_Order.client\_id = Client.id)

INNER JOIN Invoice\_Product ON Invoice\_Product.invoice\_id = Invoice.id)

INNER JOIN Product ON Invoice\_Product.product\_id = Product.id

WHERE loaded = FALSE && agreed = TRUE;

CREATE VIEW Client\_Offer AS

SELECT Client.name AS client\_name, Client.surname, Client\_Request.request, Client.email, Product.name AS product

FROM ((Client INNER JOIN Client\_Request ON Client.id = Client\_Request.client\_id)

INNER JOIN Request\_Product ON Request\_Product.request\_id = Client\_Request.id)

INNER JOIN Product ON Product.category =(SELECT category FROM Product INNER JOIN Request\_Product WHERE Product.id = Request\_Product.product\_id);

CREATE TRIGGER delete\_Client

AFTER DELETE

ON Client

FOR EACH ROW

INSERT INTO Client\_Archive(id, name, surname, hone, adress, email)

VALUES (OLD.name, OLD.surname, OLD.phone, OLD.adress, OLD.email);

DELIMITER //

CREATE PROCEDURE delete\_order(IN del\_order\_id INT)

BEGIN

SET @contract = (SELECT contract\_id FROM Client\_Order WHERE id = del\_order\_id);

DELETE FROM Client\_Order

WHERE id = del\_order\_id;

DELETE FROM Order\_Product

WHERE order\_id = del\_order\_id;

INSERT INTO Contract\_Archive(contract, begin\_date, end\_date)

(SELECT contract, begin\_date, end\_date

FROM Contract WHERE Contract.id = @contract);

DELETE FROM Contract

WHERE id = @contract;

END; //

DELIMITER ;

DELIMITER $$

CREATE TRIGGER update\_Products\_Orders

AFTER UPDATE

ON Invoice\_Product

FOR EACH ROW

BEGIN

IF NEW.loaded = TRUE THEN

UPDATE Order\_Product

SET Order\_Product.rest = Order\_Product.rest - NEW.count

WHERE Order\_Product.product\_id = NEW.product\_id;

UPDATE Product

SET Product.count = Product.count - NEW.count

WHERE Product.id = NEW.product\_id;

SET @order = (SELECT order\_id FROM Invoice RIGHT JOIN Invoice\_Product ON Invoice.id = NEW.invoice\_id GROUP BY order\_id);

SET @check = (SELECT SUM(rest) FROM order\_product WHERE order\_id = @order);

IF (@check = 0) THEN CALL delete\_order(@order); END IF;

END IF;

END$$

DELIMITER ;

DELIMITER |

CREATE TRIGGER create\_Order

AFTER UPDATE

ON Client\_Request

FOR EACH ROW

BEGIN

DECLARE x INT;

DECLARE y INT;

SET x = NEW.client\_id;

SET y = NEW.id;

IF NEW.checked = TRUE AND NEW.approved = TRUE THEN

INSERT INTO Client\_Order(client\_id,request\_id)

VALUES (x,y);

SET @lastID := LAST\_INSERT\_ID();

INSERT INTO Order\_Product(order\_id,product\_id,count,rest)

SELECT @lastID, Request\_Product.product\_id, Request\_Product.count, Request\_Product.count FROM Request\_Product

WHERE request\_id = NEW.id;

END IF;

END;

|

DELIMITER ;