USE [master]

GO

/\*\*\*\*\*\* Object: Database [Kudelka\_03] Script Date: 08.09.2019 0:00:25 \*\*\*\*\*\*/

CREATE DATABASE [Kudelka\_03]

CONTAINMENT = NONE

ON PRIMARY

( NAME = N'Kudelka\_03', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\DATA\Kudelka\_03.mdf' , SIZE = 5120KB , MAXSIZE = UNLIMITED, FILEGROWTH = 1024KB )

LOG ON

( NAME = N'Kudelka\_03\_log', FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\DATA\Kudelka\_03\_log.ldf' , SIZE = 1024KB , MAXSIZE = 2048GB , FILEGROWTH = 10%)

GO

IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))

begin

EXEC [Kudelka\_03].[dbo].[sp\_fulltext\_database] @action = 'enable'

end

GO

ALTER DATABASE [Kudelka\_03] SET ANSI\_NULL\_DEFAULT OFF

GO

ALTER DATABASE [Kudelka\_03] SET ANSI\_NULLS OFF

GO

ALTER DATABASE [Kudelka\_03] SET ANSI\_PADDING OFF

GO

ALTER DATABASE [Kudelka\_03] SET ANSI\_WARNINGS OFF

GO

ALTER DATABASE [Kudelka\_03] SET ARITHABORT OFF

GO

ALTER DATABASE [Kudelka\_03] SET AUTO\_CLOSE OFF

GO

ALTER DATABASE [Kudelka\_03] SET AUTO\_SHRINK OFF

GO

ALTER DATABASE [Kudelka\_03] SET AUTO\_UPDATE\_STATISTICS ON

GO

ALTER DATABASE [Kudelka\_03] SET CURSOR\_CLOSE\_ON\_COMMIT OFF

GO

ALTER DATABASE [Kudelka\_03] SET CURSOR\_DEFAULT GLOBAL

GO

ALTER DATABASE [Kudelka\_03] SET CONCAT\_NULL\_YIELDS\_NULL OFF

GO

ALTER DATABASE [Kudelka\_03] SET NUMERIC\_ROUNDABORT OFF

GO

ALTER DATABASE [Kudelka\_03] SET QUOTED\_IDENTIFIER OFF

GO

ALTER DATABASE [Kudelka\_03] SET RECURSIVE\_TRIGGERS OFF

GO

ALTER DATABASE [Kudelka\_03] SET DISABLE\_BROKER

GO

ALTER DATABASE [Kudelka\_03] SET AUTO\_UPDATE\_STATISTICS\_ASYNC OFF

GO

ALTER DATABASE [Kudelka\_03] SET DATE\_CORRELATION\_OPTIMIZATION OFF

GO

ALTER DATABASE [Kudelka\_03] SET TRUSTWORTHY OFF

GO

ALTER DATABASE [Kudelka\_03] SET ALLOW\_SNAPSHOT\_ISOLATION OFF

GO

ALTER DATABASE [Kudelka\_03] SET PARAMETERIZATION SIMPLE

GO

ALTER DATABASE [Kudelka\_03] SET READ\_COMMITTED\_SNAPSHOT OFF

GO

ALTER DATABASE [Kudelka\_03] SET HONOR\_BROKER\_PRIORITY OFF

GO

ALTER DATABASE [Kudelka\_03] SET RECOVERY FULL

GO

ALTER DATABASE [Kudelka\_03] SET MULTI\_USER

GO

ALTER DATABASE [Kudelka\_03] SET PAGE\_VERIFY CHECKSUM

GO

ALTER DATABASE [Kudelka\_03] SET DB\_CHAINING OFF

GO

ALTER DATABASE [Kudelka\_03] SET FILESTREAM( NON\_TRANSACTED\_ACCESS = OFF )

GO

ALTER DATABASE [Kudelka\_03] SET TARGET\_RECOVERY\_TIME = 0 SECONDS

GO

ALTER DATABASE [Kudelka\_03] SET DELAYED\_DURABILITY = DISABLED

GO

ALTER DATABASE [Kudelka\_03] SET READ\_WRITE

GO

---

--- CREATE TABLE statements

---

--DROP TABLE ORDERS;

--DROP TABLE CUSTOMERS;

--DROP TABLE SALESREPS;

--DROP TABLE OFFICES;

--DROP TABLE PRODUCTS;

CREATE TABLE PRODUCTS

(MFR\_ID CHAR(3) NOT NULL,

PRODUCT\_ID CHAR(5) NOT NULL,

DESCRIPTION VARCHAR(20) NOT NULL,

PRICE MONEY NOT NULL,

QTY\_ON\_HAND INTEGER NOT NULL,

PRIMARY KEY (MFR\_ID, PRODUCT\_ID));

CREATE TABLE OFFICES

(OFFICE INT NOT NULL,

CITY VARCHAR(15) NOT NULL,

REGION VARCHAR(10) NOT NULL,

MGR INT,

TARGET DECIMAL(9,2),

SALES DECIMAL(9,2) NOT NULL,

PRIMARY KEY (OFFICE));

CREATE TABLE SALESREPS

(EMPL\_NUM INT NOT NULL,

CHECK (EMPL\_NUM BETWEEN 101 AND 199),

NAME VARCHAR(15) NOT NULL,

AGE INTEGER,

REP\_OFFICE INTEGER,

TITLE VARCHAR(10),

HIRE\_DATE DATE NOT NULL,

MANAGER INT,

QUOTA DECIMAL(9,2),

SALES DECIMAL(9,2) NOT NULL,

PRIMARY KEY (EMPL\_NUM),

FOREIGN KEY (MANAGER) REFERENCES SALESREPS(EMPL\_NUM),

CONSTRAINT WORKSIN FOREIGN KEY (REP\_OFFICE)

REFERENCES OFFICES(OFFICE));

CREATE TABLE CUSTOMERS

(CUST\_NUM INTEGER NOT NULL,

COMPANY VARCHAR(20) NOT NULL,

CUST\_REP INTEGER,

CREDIT\_LIMIT DECIMAL(9,2),

PRIMARY KEY (CUST\_NUM),

CONSTRAINT HASREP FOREIGN KEY (CUST\_REP)

REFERENCES SALESREPS(EMPL\_NUM));

CREATE TABLE ORDERS

(ORDER\_NUM INTEGER NOT NULL,

CHECK (ORDER\_NUM > 100000),

ORDER\_DATE DATE NOT NULL,

CUST INTEGER NOT NULL,

REP INTEGER,

MFR CHAR(3) NOT NULL,

PRODUCT CHAR(5) NOT NULL,

QTY INTEGER NOT NULL,

AMOUNT DECIMAL(9,2) NOT NULL,

PRIMARY KEY (ORDER\_NUM),

CONSTRAINT PLACEDBY FOREIGN KEY (CUST)

REFERENCES CUSTOMERS(CUST\_NUM)

ON DELETE CASCADE,

CONSTRAINT TAKENBY FOREIGN KEY (REP)

REFERENCES SALESREPS(EMPL\_NUM),

CONSTRAINT ISFOR FOREIGN KEY (MFR, PRODUCT)

REFERENCES PRODUCTS(MFR\_ID, PRODUCT\_ID));

ALTER TABLE OFFICES

ADD CONSTRAINT HASMGR

FOREIGN KEY (MGR) REFERENCES SALESREPS(EMPL\_NUM);

---

--- Inserts for sample schema

---

---

--- PRODUCTS

---

INSERT INTO PRODUCTS VALUES('REI','2A45C','Ratchet Link',79.00,210);

INSERT INTO PRODUCTS VALUES('ACI','4100Y','Widget Remover',2750.00,25);

INSERT INTO PRODUCTS VALUES('QSA','XK47 ','Reducer',355.00,38);

INSERT INTO PRODUCTS VALUES('BIC','41627','Plate',180.00,0);

INSERT INTO PRODUCTS VALUES('IMM','779C ','900-LB Brace',1875.00,9);

INSERT INTO PRODUCTS VALUES('ACI','41003','Size 3 Widget',107.00,207);

INSERT INTO PRODUCTS VALUES('ACI','41004','Size 4 Widget',117.00,139);

INSERT INTO PRODUCTS VALUES('BIC','41003','Handle',652.00,3);

INSERT INTO PRODUCTS VALUES('IMM','887P ','Brace Pin',250.00,24);

INSERT INTO PRODUCTS VALUES('QSA','XK48 ','Reducer',134.00,203);

INSERT INTO PRODUCTS VALUES('REI','2A44L','Left Hinge',4500.00,12);

INSERT INTO PRODUCTS VALUES('FEA','112 ','Housing',148.00,115);

INSERT INTO PRODUCTS VALUES('IMM','887H ','Brace Holder',54.00,223);

INSERT INTO PRODUCTS VALUES('BIC','41089','Retainer',225.00,78);

INSERT INTO PRODUCTS VALUES('ACI','41001','Size 1 Wiget',55.00,277);

INSERT INTO PRODUCTS VALUES('IMM','775C ','500-lb Brace',1425.00,5);

INSERT INTO PRODUCTS VALUES('ACI','4100Z','Widget Installer',2500.00,28);

INSERT INTO PRODUCTS VALUES('QSA','XK48A','Reducer',177.00,37);

INSERT INTO PRODUCTS VALUES('ACI','41002','Size 2 Widget',76.00,167);

INSERT INTO PRODUCTS VALUES('REI','2A44R','Right Hinge',4500.00,12);

INSERT INTO PRODUCTS VALUES('IMM','773C ','300-lb Brace',975.00,28);

INSERT INTO PRODUCTS VALUES('ACI','4100X','Widget Adjuster',25.00,37);

INSERT INTO PRODUCTS VALUES('FEA','114 ','Motor Mount',243.00,15);

INSERT INTO PRODUCTS VALUES('IMM','887X ','Brace Retainer',475.00,32);

INSERT INTO PRODUCTS VALUES('REI','2A44G','Hinge Pin',350.00,14);

---

--- OFFICES

---

INSERT INTO OFFICES VALUES(22,'Denver','Western',null,300000.00,186042.00);

INSERT INTO OFFICES VALUES(11,'New York','Eastern',null,575000.00,692637.00);

INSERT INTO OFFICES VALUES(12,'Chicago','Eastern',null,800000.00,735042.00);

INSERT INTO OFFICES VALUES(13,'Atlanta','Eastern',null,350000.00,367911.00);

INSERT INTO OFFICES VALUES(21,'Los Angeles','Western',null,725000.00,835915.00);

---

--- SALESREPS

---

INSERT INTO SALESREPS VALUES (106,'Sam Clark',52,11,'VP Sales','2006-06-14',null,275000.00,299912.00);

INSERT INTO SALESREPS VALUES (109,'Mary Jones',31,11,'Sales Rep','2007-10-12',106,300000.00,392725.00);

INSERT INTO SALESREPS VALUES (104,'Bob Smith',33,12,'Sales Mgr','2005-05-19',106,200000.00,142594.00);

INSERT INTO SALESREPS VALUES (108,'Larry Fitch',62,21,'Sales Mgr','2007-10-12',106,350000.00,361865.00);

INSERT INTO SALESREPS VALUES (105,'Bill Adams',37,13,'Sales Rep','2006-02-12',104,350000.00,367911.00);

INSERT INTO SALESREPS VALUES (102,'Sue Smith',48,21,'Sales Rep','2004-12-10',108,350000.00,474050.00);

INSERT INTO SALESREPS VALUES (101,'Dan Roberts',45,12,'Sales Rep','2004-10-20',104,300000.00,305673.00);

INSERT INTO SALESREPS VALUES (110,'Tom Snyder',41,null,'Sales Rep','2008-01-13',101,null,75985.00);

INSERT INTO SALESREPS VALUES (103,'Paul Cruz',29,12,'Sales Rep','2005-03-01',104,275000.00,286775.00);

INSERT INTO SALESREPS VALUES (107,'Nancy Angelli',49,22,'Sales Rep','2006-11-14',108,300000.00,186042.00);

---

--- OFFICE MANAGERS

---

UPDATE OFFICES SET MGR=108 WHERE OFFICE=22;

UPDATE OFFICES SET MGR=106 WHERE OFFICE=11;

UPDATE OFFICES SET MGR=104 WHERE OFFICE=12;

UPDATE OFFICES SET MGR=105 WHERE OFFICE=13;

UPDATE OFFICES SET MGR=108 WHERE OFFICE=21;

---

--- CUSTOMERS

---

INSERT INTO CUSTOMERS VALUES(2111,'JCP Inc.',103,50000.00);

INSERT INTO CUSTOMERS VALUES(2102,'First Corp.',101,65000.00);

INSERT INTO CUSTOMERS VALUES(2103,'Acme Mfg.',105,50000.00);

INSERT INTO CUSTOMERS VALUES(2123,'Carter \& Sons',102,40000.00);

INSERT INTO CUSTOMERS VALUES(2107,'Ace International',110,35000.00);

INSERT INTO CUSTOMERS VALUES(2115,'Smithson Corp.',101,20000.00);

INSERT INTO CUSTOMERS VALUES(2101,'Jones Mfg.',106,65000.00);

INSERT INTO CUSTOMERS VALUES(2112,'Zetacorp',108,50000.00);

INSERT INTO CUSTOMERS VALUES(2121,'QMA Assoc.',103,45000.00);

INSERT INTO CUSTOMERS VALUES(2114,'Orion Corp.',102,20000.00);

INSERT INTO CUSTOMERS VALUES(2124,'Peter Brothers',107,40000.00);

INSERT INTO CUSTOMERS VALUES(2108,'Holm \& Landis',109,55000.00);

INSERT INTO CUSTOMERS VALUES(2117,'J.P. Sinclair',106,35000.00);

INSERT INTO CUSTOMERS VALUES(2122,'Three Way Lines',105,30000.00);

INSERT INTO CUSTOMERS VALUES(2120,'Rico Enterprises',102,50000.00);

INSERT INTO CUSTOMERS VALUES(2106,'Fred Lewis Corp.',102,65000.00);

INSERT INTO CUSTOMERS VALUES(2119,'Solomon Inc.',109,25000.00);

INSERT INTO CUSTOMERS VALUES(2118,'Midwest Systems',108,60000.00);

INSERT INTO CUSTOMERS VALUES(2113,'Ian \& Schmidt',104,20000.00);

INSERT INTO CUSTOMERS VALUES(2109,'Chen Associates',103,25000.00);

INSERT INTO CUSTOMERS VALUES(2105,'AAA Investments',101,45000.00);

---

--- ORDERS

---

INSERT INTO ORDERS VALUES (112961,'2007-12-17',2117,106,'REI','2A44L',7,31500.00);

INSERT INTO ORDERS VALUES (113012,'2008-01-11',2111,105,'ACI','41003',35,3745.00);

INSERT INTO ORDERS VALUES (112989,'2008-01-03',2101,106,'FEA','114',6,1458.00);

INSERT INTO ORDERS VALUES (113051,'2008-02-10',2118,108,'QSA','XK47',4,1420.00);

INSERT INTO ORDERS VALUES (112968,'2007-10-12',2102,101,'ACI','41004',34,3978.00);

INSERT INTO ORDERS VALUES (113036,'2008-01-30',2107,110,'ACI','4100Z',9,22500.00);

INSERT INTO ORDERS VALUES (113045,'2008-02-02',2112,108,'REI','2A44R',10,45000.00);

INSERT INTO ORDERS VALUES (112963,'2007-12-17',2103,105,'ACI','41004',28,3276.00);

INSERT INTO ORDERS VALUES (113013,'2008-01-14',2118,108,'BIC','41003',1,652.00);

INSERT INTO ORDERS VALUES (113058,'2008-02-23',2108,109,'FEA','112',10,1480.00);

INSERT INTO ORDERS VALUES (112997,'2008-01-08',2124,107,'BIC','41003',1,652.00);

INSERT INTO ORDERS VALUES (112983,'2007-12-27',2103,105,'ACI','41004',6,702.00);

INSERT INTO ORDERS VALUES (113024,'2008-01-20',2114,108,'QSA','XK47',20,7100.00);

INSERT INTO ORDERS VALUES (113062,'2008-02-24',2124,107,'FEA','114',10,2430.00);

INSERT INTO ORDERS VALUES (112979,'2007-10-12',2114,102,'ACI','4100Z',6,15000.00);

INSERT INTO ORDERS VALUES (113027,'2008-01-22',2103,105,'ACI','41002',54,4104.00);

INSERT INTO ORDERS VALUES (113007,'2008-01-08',2112,108,'IMM','773C',3,2925.00);

INSERT INTO ORDERS VALUES (113069,'2008-03-02',2109,107,'IMM','775C',22,31350.00);

INSERT INTO ORDERS VALUES (113034,'2008-01-29',2107,110,'REI','2A45C',8,632.00);

INSERT INTO ORDERS VALUES (112992,'2007-11-04',2118,108,'ACI','41002',10,760.00);

INSERT INTO ORDERS VALUES (112975,'2007-10-12',2111,103,'REI','2A44G',6,2100.00);

INSERT INTO ORDERS VALUES (113055,'2008-02-15',2108,101,'ACI','4100X',6,150.00);

INSERT INTO ORDERS VALUES (113048,'2008-02-10',2120,102,'IMM','779C',2,3750.00);

INSERT INTO ORDERS VALUES (112993,'2007-01-04',2106,102,'REI','2A45C',24,1896.00);

INSERT INTO ORDERS VALUES (113065,'2008-02-27',2106,102,'QSA','XK47',6,2130.00);

INSERT INTO ORDERS VALUES (113003,'2008-01-25',2108,109,'IMM','779C',3,5625.00);

INSERT INTO ORDERS VALUES (113049,'2008-02-10',2118,108,'QSA','XK47',2,776.00);

INSERT INTO ORDERS VALUES (112987,'2007-12-31',2103,105,'ACI','4100Y',11,27500.00);

INSERT INTO ORDERS VALUES (113057,'2008-02-18',2111,103,'ACI','4100X',24,600.00);

INSERT INTO ORDERS VALUES (113042,'2008-02-20',2113,101,'REI','2A44R',5,22500.00);

--- lab\_03

---3.1 Выбрать фамилии и даты найма всех сотрудников.

SELECT NAME, HIRE\_DATE from SALESREPS;

---3.2 Выбрать все заказы, выполненные после определенной даты

SELECT \* FROM ORDERS WHERE ORDER\_DATE > '2008-01-01';

---3.3. Выбрать все офисы из определенного региона и управляемые определенным сотрудником.

SELECT \* FROM OFFICES WHERE REGION = 'WESTERN' and MGR = 108;

---3.4. Выбрать заказы, сумма которых больше определенного значения.

SELECT \* FROM ORDERS WHERE AMOUNT > 10000;

---3.5. Выбрать заказы определенного покупателя.

SELECT \* FROM ORDERS WHERE CUST = 2109;

---3.6. Выбрать заказы, сделанные в определенный период.

SELECT \* FROM ORDERS WHERE ORDER\_DATE = '2007-10-12';

---3.7. Выбрать офисы из 12, 13 и 21 региона.

SELECT \* FROM OFFICES WHERE OFFICE = 12 OR OFFICE = 13 OR OFFICE =21;

---3.8. Выбрать сотрудника, у которого нет менеджера (самого главного).

SELECT \* FROM SALESREPS WHERE MANAGER IS NULL;

---3.9. Выбрать офисы из региона, который начинается на East.

SELECT \* FROM OFFICES WHERE REGION LIKE 'East%';

---3.10. Выбрать продукты с ценой больше определенного значения и отсортировать в порядке убывания цены.

SELECT \* FROM PRODUCTS WHERE PRICE >1000 ORDER BY PRICE DESC;

---3.11. Выбрать фамилии и даты найма всех сотрудников и отсортировать по возрасту.

SELECT NAME, HIRE\_DATE FROM SALESREPS ORDER BY AGE;

---3.12. Выбрать все заказы и отсортировать вначале по стоиомости по убыванию, а затем по количеству заказанного по возрастанию.

SELECT \* FROM ORDERS ORDER BY AMOUNT DESC, QTY ASC;

---3.13. Выбрать 5 самых дорогих товаров.

SELECT TOP 5 \* FROM PRODUCTS

ORDER BY PRICE DESC;

---3.14. Выбрать 3 самых молодых сотрудников.

SELECT TOP 3 \* FROM SALESREPS

ORDER BY AGE ASC;

---3.15. Выбрать 20% самых дорогих заказов.

SELECT TOP 20 PERCENT \* FROM ORDERS

ORDER BY AMOUNT DESC;

---3.16. Выбрать 11 покупателей с самым высоким кредитным лимитом.

SELECT TOP 11 WITH TIES \* FROM CUSTOMERS

ORDER BY CREDIT\_LIMIT DESC;

---3.17. Выбрать сотрудников с 4 по 7, отсортированных по дате найма.

SELECT \* FROM SALESREPS ORDER BY HIRE\_DATE OFFSET 3 ROWS FETCH NEXT 4 ROWS ONLY;

---3.18. Выбрать сотрудников с 4 по 7, отсортированных по возрасту и тех, кто с ними одного возраста.

SELECT \* FROM SALESREPS ORDER BY AGE OFFSET 3 ROWS FETCH NEXT 4 ROWS ONLY;

---3.19. Выбрать уникальные товары в заказах.

SELECT DISTINCT PRODUCT FROM ORDERS;

---3.20. Подсчитать количество заказов для каждого покупателя.

SELECT CUST, COUNT(ORDER\_NUM) AS nnn FROM ORDERS GROUP BY CUST;

---3.21. Подсчитать итоговую сумму заказа для каждого покупателя.

SELECT CUST, SUM(AMOUNT) AS SSS FROM ORDERS GROUP BY CUST;

---3.22. Подсчитать среднюю цену заказа для каждого сотрудника.

SELECT REP, AVG(AMOUNT) AS RRR FROM ORDERS GROUP BY REP;

---3.23. Найти сотрудников, у которых есть заказ стоимости выше определенного значения.

SELECT REP, AMOUNT FROM ORDERS WHERE AMOUNT > 10000;

---3.24. Найти количество продуктов для каждого производителя.

SELECT MFR\_ID, COUNT(PRODUCT\_ID) AS PPP FROM PRODUCTS GROUP BY MFR\_ID;

---3.25. Найти самый дорогой товар каждого производителя.

SELECT MFR\_ID, MAX(PRICE) AS PRI FROM PRODUCTS GROUP BY MFR\_ID;

---3.26. Найти покупателей и их заказы (в результирующем наборе должны быть: наименование покупателя, наименование товара, производитель, количество и итоговая сумма).

SELECT COMPANY, PRODUCT, MFR, QTY, AMOUNT

FROM ORDERS O JOIN CUSTOMERS C

ON O.CUST = C.CUST\_NUM;

---3.27. Найти всех покупателей и их заказы.

SELECT C.COMPANY, O.ORDER\_NUM

FROM ORDERS O RIGHT JOIN CUSTOMERS C

ON O.CUST = C.CUST\_NUM;

---3.28. Найти покупателей, у которых нет заказов.

INSERT INTO CUSTOMERS VALUES(4000,'TATSIANA',106,45000.00);

SELECT \* FROM CUSTOMERS WHERE CUSTOMERS.CUST\_NUM NOT IN

(SELECT ORDERS.CUST FROM ORDERS);

---3.29. Найти покупателей, у которых есть заказы в определенный период.

SELECT CUSTOMERS.COMPANY, ORDERS.ORDER\_DATE FROM CUSTOMERS JOIN ORDERS ON ORDERS.CUST = CUSTOMERS.CUST\_NUM WHERE ORDER\_DATE BETWEEN '2007-12-01' AND '2007-12-31';

---3.30. Найти покупателей, у которых есть заказы выше определенной суммы.

SELECT CUSTOMERS.COMPANY, ORDERS.AMOUNT FROM CUSTOMERS JOIN ORDERS ON ORDERS.CUST = CUSTOMERS.CUST\_NUM WHERE ORDERS.AMOUNT > 30000;

---3.31. Найти заказы, которые оформляли менеджеры из региона EAST.

SELECT \* FROM ORDERS WHERE REP IN

(SELECT SALESREPS.EMPL\_NUM FROM OFFICES JOIN SALESREPS ON OFFICES.OFFICE = SALESREPS.REP\_OFFICE WHERE OFFICES.REGION = 'Eastern');

---3.32. Найти товары, которые купили покупатели с кредитным лимитом больше 40000.

SELECT \* FROM PRODUCTS WHERE PRODUCT\_ID IN

(SELECT ORDERS.PRODUCT FROM ORDERS JOIN CUSTOMERS ON CUSTOMERS.CUST\_NUM = ORDERS.CUST WHERE CUSTOMERS.CREDIT\_LIMIT > 40000);

---3.33. Найти всех сотрудников из региона EAST и все их заказы.

SELECT \* FROM ORDERS, SALESREPS WHERE SALESREPS.EMPL\_NUM = ORDERS.REP AND ORDERS.REP IN

(SELECT SALESREPS.EMPL\_NUM FROM OFFICES JOIN SALESREPS ON OFFICES.OFFICE = SALESREPS.REP\_OFFICE WHERE OFFICES.REGION = 'Eastern');

---3.34. Найти сотрудников, которые не оформили ни одного заказа.

SELECT \* FROM SALESREPS WHERE SALESREPS.EMPL\_NUM NOT IN

(SELECT ORDERS.REP FROM ORDERS);

---3.35. Найти сотрудников одного возраста.

INSERT INTO SALESREPS VALUES (111,'Rembo', 33, 13, 'Sales Rep', '2019-02-12', 104, 50000.00, 67911.00)

SELECT \* FROM SALESREPS WHERE AGE =

(SELECT AGE FROM SALESREPS GROUP BY AGE HAVING COUNT(\*) > 1);

---4. Поместить результирующие наборы из запроса 3.30 в локальную временную таблицу.

SELECT CUSTOMERS.COMPANY, ORDERS.AMOUNT INTO #MYTABLE FROM CUSTOMERS JOIN ORDERS ON ORDERS.CUST = CUSTOMERS.CUST\_NUM WHERE ORDERS.AMOUNT > 30000;

---5. Просмотреть данные из локальной временной таблицы.

SELECT \* FROM #MYTABLE;

---6. Поместить результирующие наборы из запроса 3.31 в глобальную временную таблицу.

SELECT \* INTO ##MYTABLE\_3\_31 FROM ORDERS WHERE REP IN

(SELECT SALESREPS.EMPL\_NUM FROM OFFICES JOIN SALESREPS ON OFFICES.OFFICE = SALESREPS.REP\_OFFICE WHERE OFFICES.REGION = 'Eastern');

---7. Просмотреть данные из глобальных временных таблиц.

SELECT \* FROM ##MYTABLE\_3\_31;

SELECT DISTINCT C.COMPANY, M.MFR

FROM ORDERS M JOIN CUSTOMERS C

ON M.CUST = C.CUST\_NUM

ORDER BY C.COMPANY;

Select \* from OFFICES;

SELECT T.CITY, count(MFR) AS h

from ORDERS O

JOIN SALESREPS S ON O.REP = S.EMPL\_NUM

JOIN OFFICES T ON S.REP\_OFFICE= T.OFFICE

GROUP BY T.CITY

having count(MFR)>1;

SELECT T.CITY, count(DISTINCT MFR) AS K

from ORDERS O

JOIN SALESREPS S ON O.REP = S.EMPL\_NUM

JOIN OFFICES T ON S.REP\_OFFICE= T.OFFICE

GROUP BY T.CITY

having count(MFR)>1;

SELECT MFR\_ID, COUNT(PRODUCT\_ID) AS nnn, AVG(PRICE) AS RRR FROM PRODUCTS GROUP BY MFR\_ID;