

Практика по базам данных
ОТЧЕТ

Долгополова Мария
371 группа

Предметная область: «Кофейная компания»

Оглавление

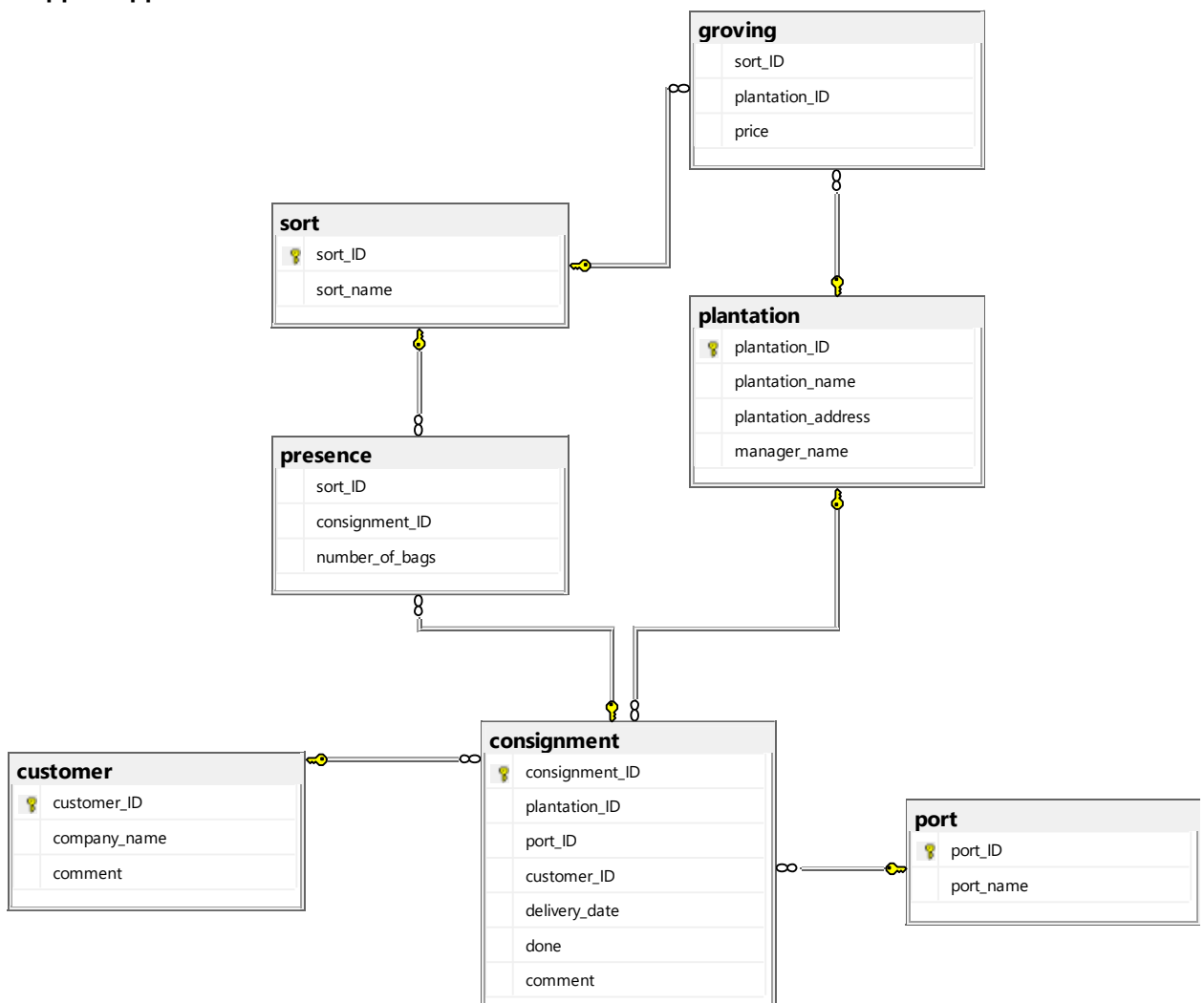
ОПИСАНИЕ СИСТЕМЫ	2
Требования.....	2
Модель данных	2
Функциональность	3
Серверная часть.....	3
Клиентская часть	4
СКРИПТЫ	5
Серверная часть	5
Хранимые процедуры и функции	5
Триггеры	5
Представления	6
Клиентская часть	7
Создание и заполнение базы данных	9

ОПИСАНИЕ СИСТЕМЫ

Требования

Кофейная компания выращивает, перевозит и продает перекупщикам кофе. Компания владеет несколькими плантациями в Южной Америке (о плантации известны адрес, данные об управляющем, сорта выращиваемого кофе, цена продукции). Кофе упаковывается на плантации в стандартные мешки и отправляется в порт. Заказчиком партии является какая-либо компания, фасующая и продающая кофе-продукты. Нужно отслеживать перемещение партий кофе с плантаций в порты.

Модель данных



Функциональность

Серверная часть

Хранимые процедуры/функции	Реализация	Комментарии
Регистрация нового заказчика		(название компании)
Добавление нового сорта кофе		(название сорта, плантация, цена)
Регистрация новой плантации		(название плантации, адрес, управляющий)
Регистрация нового порта		(название порта)
Регистрация сорта кофе на плантации		(сорт, плантация, цена)
Добавление нового заказа (новой партии)		(плантация, порт, заказчик, дата поставки) Проверка того, что мы работаем с этим портом и этой плантацией
Уточнение информации о заказе	Clarification()	(заказ, сорт кофе, количество мешков) Проверка того, что сорт кофе выращивается на плантации, указанной в заказе
Выполнение заказа	Execution()	(заказ, комментарий)
Удаление заказа		(заказ)
Количество заказов от заказчика за месяц		(заказчик)
Количество заказов в порт		(порт)
Вычисление стоимости заказа		(заказ)
Поиск «любимой» плантации заказчика (плантация, кофе с которой заказывает чаще всего)	FavoritePlantation()	(заказчик)

Триггеры	Реализация	Комментарии
Запрет на удаление заказчика	Tr_del_customer	Для сохранения клиентской базы
Можно удалить только выполненные заказы		
После 3-го заказа заказчик становится постоянным клиентом	Tr_ins_consignment_regular	В комментариях у заказчика «REGULAR CUSTOMER»
Деактивация вместо удаления заказчика	Tr_del_customer_DEACTIVATED	Для сохранения клиентской базы В комментариях у заказчика «DEACTIVATED»
Контроль повторного добавления сорта		Нельзя добавить сорт с названием, которое уже зарегистрировано
Добавление комментария «New customer» при добавлении нового клиента		

Представления	Реализация	Комментарии
Все заказы	V_OrderList	все заказы (id, плантация, порт, заказчик, дата, количество сортов)
Плантации с адресами и управляющими		
Количество выполненных заказов у заказчиков	V_CustomerRating	(id, название компании, количество заказов)
Заказы в порт 'Guayaquil'		

Выполненные заказы		
Просроченные заказы		
Средняя стоимость сортов по плантациям	V_Price	(id, название, avg)

Клиентская часть

Экранные формы основные	дополнительные	Реализация (запрос)	Что здесь можно использовать из серверной части
Список плантаций	Получение информации о сортах	(01) Данные о плантациях с адресом и выращиваемыми сортами	
	Добавить		
	Удалить		
	Зарегистрировать сорта кофе		
	Фильтры	(02) Плантации, на которых выращивается больше 2 сортов (03) Плантации, на которых ничего не выращивается	
Список заказчиков	Добавить		
	Удалить		Tr_del_customer Tr_del_customer_DEACTIVATED
	Фильтры	(04) 3 заказчика, у которых больше всего выполненных заказов (05) Заказчики, являющиеся постоянными клиентами	CustomerRating
	Добавить комментарий		
Список сортов кофе	Добавить		
	Удалить		
	Фильтры	(06) Кофе, стоимость которого выше средней (07) Сорта кофе, в названии которых есть Бразилия	Price
Список портов	Добавить		
	Удалить		
	Фильтры	(08) Список всех заказов, доставляемых в порт, в который доставляется заказ с id=1 (09) Порты, в которые есть заказы для Starbucks	
Список заказов	Фильтр	(10) Просроченные заказы	OrderList
	Добавить		Tr_ins_consignment_regular
	Выполнить		Execution()
	Добавить информацию о заказе		Clarification()
	Удалить		
Служебные запросы		(11) Все комментарии по заказам и клиентам (12) Список всех портов и плантаций	

СКРИПТЫ

Серверная часть

Хранимые процедуры и функции

```
-- Добавление сорта в заказ
CREATE PROCEDURE Clarification
@consignment_id AS INT,
@sort_id AS INT,
@number_of_bags AS INT
AS
BEGIN
    DECLARE @plantation INTEGER;
    SELECT @plantation = consignment.plantation_ID FROM consignment WHERE
consignment_ID = @consignment_id
    IF (SELECT COUNT(*) FROM groving where groving.plantation_ID=@plantation AND
groving.sort_ID = @sort_id)>0
        INSERT INTO presence VALUES (@sort_id, @consignment_id, @number_of_bags)
END;
-- Пример вызова процедуры:
-- EXECUTE Clarification @consignment_id=4, @sort_id = 9, @number_of_bags=10
-- DROP PROCEDURE Clarification;

-- Выполнить заказ
CREATE PROCEDURE Execution
@consignment_id AS INT,
@comment AS NVARCHAR(200)
AS
BEGIN
    DECLARE @var_new_person_id integer;
    DECLARE @var_new_owner_id integer;
    UPDATE consignment SET done = 1, comment = @comment
        WHERE consignment_ID = @consignment_id
END;
-- Пример вызова процедуры:
-- EXECUTE Execution @consignment_id = 10, @comment = '))))))'
-- DROP PROCEDURE Execution;

-- Поиск любимой плантации заказчика
CREATE FUNCTION FavoritePlantation(@customer_id INT)
RETURNS NVARCHAR(20)
BEGIN
    DECLARE @id integer;
    DECLARE @name NVARCHAR(20);
    SELECT @id = plantation_ID FROM consignment
        WHERE consignment.customer_ID = @customer_id
        GROUP BY plantation_ID
        ORDER BY COUNT(plantation_ID) DESC
    SELECT @name = plantation.plantation_name FROM plantation
        WHERE plantation.plantation_ID = @id
    RETURN @name
END;
-- Вызов:
-- SELECT dbo.FavoritePlantation(4)
-- DROP FUNCTION FavoritePlantation;
```

Триггеры

```
-- Запрет на удаление заказчика
CREATE TRIGGER TR_del_customer ON customer FOR DELETE
AS
```

```

ROLLBACK;
-- Проверка: DELETE FROM customer;
-- DROP TRIGGER TR_del_customer

-- После 3 заказа заказчик становится постоянным клиентом
CREATE TRIGGER Tr_ins_consignment_regular ON consignment FOR INSERT
AS
BEGIN
    IF (SELECT COUNT(*) FROM consignment
        WHERE consignment.customer_ID = (SELECT customer_ID FROM INSERTED)) >
2
    UPDATE customer SET comment = ' REGULAR CUSTOMER ' + comment WHERE
customer.customer_ID = (SELECT customer_ID FROM INSERTED);
    END;
-- Проверка: INSERT INTO consignment(plantation_ID, port_ID, customer_ID, delivery_date)
VALUES (4, 2, 4, '2020-10-13');
-- DROP TRIGGER Tr_ins_consignment_regular

-- Деактивация вместо удаления заказчика
CREATE TRIGGER Tr_del_customer_DEACTIVATED ON customer INSTEAD OF DELETE
AS
    UPDATE customer SET comment = ' DEACTIVATED ' + comment
        WHERE customer_ID = (SELECT customer_ID FROM DELETED);
-- Проверка: DELETE FROM customer WHERE customer_ID = 1
-- DROP TRIGGER Tr_del_customer_DEACTIVATED

```

Представления

```

-- Список заказов с количеством сортов
CREATE VIEW V_OrderList(id, done, port, plantation, customer, date, sorts)
AS
    SELECT consignment.consignment_ID, done, port.port_name,
plantation.plantation_name, customer.company_name, consignment.delivery_date,
COUNT(presence.sort_ID) AS sorts
    FROM consignment, port, presence, plantation, customer
        WHERE consignment.port_ID = port.port_ID AND consignment.plantation_ID =
plantation.plantation_ID
        AND consignment.customer_ID = customer.customer_ID AND
consignment.consignment_ID = presence.consignment_ID
        GROUP BY consignment.consignment_ID, done, port.port_name,
plantation.plantation_name, customer.company_name, consignment.delivery_date
-- DROP VIEW OrderList
-- SELECT * FROM V_OrderList

-- Заказчики с количеством выполненных заказов
CREATE VIEW V_CustomerRating (customer_id, company_name, count)
AS
    SELECT customer.customer_ID, customer.company_name,
COUNT(consignment.consignment_ID)
    FROM customer JOIN consignment ON consignment.customer_ID = customer.customer_ID
        WHERE consignment.done = 1 GROUP BY customer.customer_ID,
customer.company_name
-- DROP VIEW CustomerRating;
-- SELECT * FROM V_CustomerRating

-- Средняя цена кофе на плантациях
CREATE VIEW V_Price (plantation_id, plantation_name, avg)
AS
    SELECT plantation.plantation_ID, plantation.plantation_name, AVG(groving.price)
    FROM plantation JOIN groving ON plantation.plantation_ID = groving.plantation_ID
        GROUP BY plantation.plantation_ID, plantation.plantation_name
-- DROP VIEW Price
-- SELECT * FROM V_Price

```

Клиентская часть

```
-- (01) Данные о плантациях с адресом и выращиваемыми сортами
SELECT plantation_name, plantation_address, sort.sort_name
  FROM plantation LEFT OUTER JOIN groving ON plantation.plantation_ID =
groving.plantation_ID
  LEFT OUTER JOIN sort ON groving.sort_ID = sort.sort_ID

-- (02) Плантации, на которых выращивается больше 2 сортов
SELECT plantation_name, COUNT(*)
  FROM plantation JOIN groving ON plantation.plantation_ID = groving.plantation_ID
  GROUP BY plantation_name
  HAVING COUNT(*) > 2

-- (03) Плантации, на которых ничего не выращивается
SELECT plantation_name
  FROM plantation
  WHERE NOT EXISTS (SELECT * FROM groving WHERE plantation.plantation_ID =
groving.plantation_ID)

-- (04) 3 заказчика, у которых больше всего выполненных заказов
SELECT TOP(3) company_name, count FROM CustomerRating
  ORDER BY count DESC

-- (05) Заказчики, являющиеся постоянными клиентами
SELECT company_name FROM customer
  WHERE comment LIKE '%REGULAR%'

-- (06) Кофе, стоимость которого выше средней
SELECT sort.sort_name, plantation.plantation_name, groving.price
  FROM sort, plantation, groving
  WHERE sort.sort_ID = groving.sort_ID AND plantation.plantation_ID =
groving.plantation_ID AND
  price > (SELECT AVG(price) FROM groving)

-- (07) Сорта кофе, в названии которых есть Бразилия
SELECT sort_name FROM sort
  WHERE sort.sort_name LIKE '%Бразилия%'

-- (08) Список всех заказов, доставляемых в порт, в который доставляется заказ с id=1
SELECT consignment_ID, port_ID
  FROM consignment
  WHERE port_ID = (SELECT port_ID FROM consignment WHERE consignment_ID = 1)

-- (09) Порты, в которые есть заказы для Starbucks
SELECT port.port_name
  FROM port,
  (SELECT DISTINCT consignment.port_ID FROM consignment, customer
  WHERE consignment.customer_ID = customer.customer_ID AND
customer.company_name = 'Starbucks') temp
  WHERE port.port_ID = temp.port_ID

-- (10) Просроченные заказы
SELECT * FROM OrderList
  WHERE date < GETDATE() AND done = 0

-- (11) Все комментарии по заказам и клиентам
SELECT customer_ID AS id, comment, 'customer' FROM customer WHERE comment != ''
UNION ALL
SELECT consignment_ID AS id, comment, 'consinnment' FROM consignment WHERE comment != ''

-- (12) Список всех портов и плантаций
SELECT plantation_ID AS id, plantation_name AS name, 'plantation' FROM plantation
```

```
UNION ALL  
SELECT port_ID, port_name, 'port' FROM port
```


Создание и заполнение базы данных

```
CREATE DATABASE coffee_db;
GO
USE coffee_db;

CREATE TABLE sort (
    sort_ID INT IDENTITY(1,1) PRIMARY KEY,
    sort_name NVARCHAR(50) NOT NULL,
)

CREATE TABLE plantation (
    plantation_ID INT IDENTITY(1,1) PRIMARY KEY,
    plantation_name NVARCHAR(20) NOT NULL,
    plantation_address NVARCHAR(100) NOT NULL,
    manager_name NVARCHAR(50) NOT NULL
)

CREATE TABLE groving (
    sort_ID INT NOT NULL,
    plantation_ID INT NOT NULL,
    price money NOT NULL
    FOREIGN KEY (sort_ID) REFERENCES sort(sort_ID)
        ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY (plantation_ID) REFERENCES plantation(plantation_ID)
        ON UPDATE CASCADE ON DELETE CASCADE,
    UNIQUE(sort_ID, plantation_ID)
)

CREATE TABLE customer (
    customer_ID INT IDENTITY(1,1) PRIMARY KEY,
    company_name NVARCHAR(50) NOT NULL,
    comment NVARCHAR(200) DEFAULT ''
)

CREATE TABLE port (
    port_ID INT IDENTITY(1,1) PRIMARY KEY,
    port_name NVARCHAR(50) NOT NULL,
)

CREATE TABLE consignment (
    consignment_ID INT IDENTITY(1,1) PRIMARY KEY,
    plantation_ID INT NOT NULL,
    port_ID INT NOT NULL,
    customer_ID INT NOT NULL,
    delivery_date DATE NOT NULL,
    done INT CHECK(done >= 0) DEFAULT 0,
    comment NVARCHAR(200) DEFAULT ''
    FOREIGN KEY (plantation_ID) REFERENCES plantation(plantation_ID)
        ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY (port_ID) REFERENCES port(port_ID)
        ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY (customer_ID) REFERENCES customer(customer_ID)
        ON UPDATE CASCADE ON DELETE CASCADE,
    UNIQUE(plantation_ID, port_ID, customer_ID, delivery_date)
)

CREATE TABLE presence (
    sort_ID INT NOT NULL,
    consignment_ID INT NOT NULL,
    number_of_bags INT NOT NULL
    UNIQUE(sort_ID, consignment_ID)
)
```

```

ALTER TABLE presence ADD CONSTRAINT FK_presence_sort
    FOREIGN KEY (sort_ID)
    REFERENCES sort(sort_ID)
    ON UPDATE CASCADE ON DELETE CASCADE

ALTER TABLE presence ADD CONSTRAINT FK_presence_consignment
    FOREIGN KEY (consignment_ID)
    REFERENCES consignment(consignment_ID)
    ON UPDATE CASCADE ON DELETE CASCADE

INSERT INTO sort(sort_name) VALUES ('Арабика Боливия'), ('Арабика Бразилия Сантос'),
('Арабика Бразилия PB'),
('Арабика Бразилия Желтый Бурбон'), ('Арабика Эквадор Галапагос'), ('Арабика
Колумбия Супremo'),
('Арабика Колумбия без кофеина'), ('Арабика Колумбия Марагоджип'), ('Арабика
Колумбия Эксельсо')

INSERT INTO plantation(plantation_name, plantation_address, manager_name) VALUES
('Bolivia Coffee', 'Bolivia', 'Williams A.'),
('Premium Coffee', 'Brazil', 'Jones H.'), ('Luxury', 'Ecuador', 'Brown W.'),
('Colombia', 'Colombia', 'Johnson T.'),
('Colombia_2', 'Colombia', 'Johnson T.')

INSERT INTO groving VALUES (1, 1, 100), (2, 2, 150), (3, 2, 150), (4, 2, 120), (5, 3,
300), (6, 4, 200), (7, 4, 80), (8, 4, 110), (9, 4, 120)

INSERT INTO customer(company_name) VALUES ('Starbucks'), ('Tchibo'), ('Costa Coffee'),
('McCafe'), ('Segafredo'), ('Caffe Nero'),
('Caffe Ritazza'), ('Caffe Revive'), ('Hema Koffiebuffet'), ('Кофе Хауз'), ('BB's
Coffee & Muffins'), ('Coffeeright')

INSERT INTO port(port_name) VALUES ('Guayaquil'), ('Pecem'), ('Cartagena'), ('Aratu'),
('Belem')

INSERT INTO consignment(plantation_ID, port_ID, customer_ID, delivery_date, done) VALUES
(1, 1, 1, '2019-10-10', 0), (2, 2, 2, '2019-10-11', 1),
(3, 1, 3, '2019-10-12', 1), (4, 3, 4, '2019-11-20', 0), (2, 4, 5, '2019-11-12',
1), (2, 5, 6, '2019-10-10', 1),
(1, 1, 7, '2019-11-12', 1), (2, 2, 8, '2019-11-12', 1), (3, 1, 9, '2019-11-20',
0), (4, 3, 10, '2019-12-10', 0), (2, 4, 1, '2019-12-01', 0),
(2, 5, 12, '2019-12-02', 0), (2, 2, 12, '2019-12-30', 0), (3, 1, 10, '2019-12-20',
0), (4, 3, 2, '2019-12-10', 0), (2, 4, 4, '2019-12-13', 0),
(2, 2, 6, '2020-01-20', 0), (3, 1, 4, '2020-01-13', 0), (3, 1, 1, '2020-01-10', 0),
(4, 3, 9, '2020-02-03', 0), (2, 4, 3, '2020-03-10', 0),
(2, 5, 4, '2020-04-10', 0), (2, 2, 5, '2020-04-11', 0)

INSERT INTO presence VALUES (1, 1, 10), (2, 2, 20), (3, 3, 30), (4, 4, 40), (5, 5, 50),
(1, 2, 5), (1, 3, 6), (1, 10, 10),
(1, 12, 10), (1, 19, 5),
(2, 23, 30), (2, 4, 20), (2, 6, 15), (2, 8, 15), (2, 9, 10), (2, 14, 20), (2, 15,
5), (2, 17, 2), (2, 20, 15), (2, 21, 3),
(3, 7, 30), (3, 11, 20), (3, 6, 15), (8, 12, 15), (3, 9, 10), (9, 14, 20), (4, 1,
5), (4, 13, 2), (4, 22, 15), (5, 4, 3),
(5, 23, 30), (6, 4, 20), (6, 6, 15), (6, 12, 15), (6, 21, 10), (7, 16, 20), (7,
17, 5), (7, 18, 2), (8, 19, 15), (8, 20, 3),
(8, 21, 2), (8, 22, 20), (8, 23, 15), (9, 8, 15), (9, 9, 10), (9, 15, 20), (9, 16,
5), (9, 17, 2), (9, 20, 15), (9, 21, 3)

```

Создание индексов

```
CREATE INDEX idx_delivery_date ON consignment(delivery_date);
```

```
CREATE UNIQUE INDEX idx_plantation_name_address ON plantation(plantation_name,  
plantation_address);  
CREATE INDEX idx_company_name ON customer (company_name);
```

Удаление таблиц

```
/*  
DROP TABLE IF EXISTS groving  
DROP TABLE IF EXISTS presence  
DROP TABLE IF EXISTS consignment  
DROP TABLE IF EXISTS plantation  
DROP TABLE IF EXISTS sort  
DROP TABLE IF EXISTS port  
DROP TABLE IF EXISTS customer  
*/
```