Практика по базам данных

ОТЧЕТ

Гогина Олеся

371 группа

Предметная область:

«Прокат сноубордов»

Содержание

ОПИСАНИЕ СИСТЕМЫ

Требования

Для предприятия по прокату сноубордов нужно разработать систему учета данных о спортивном инвентаре.

Инвентарь представляет собой предметы 2х видов:

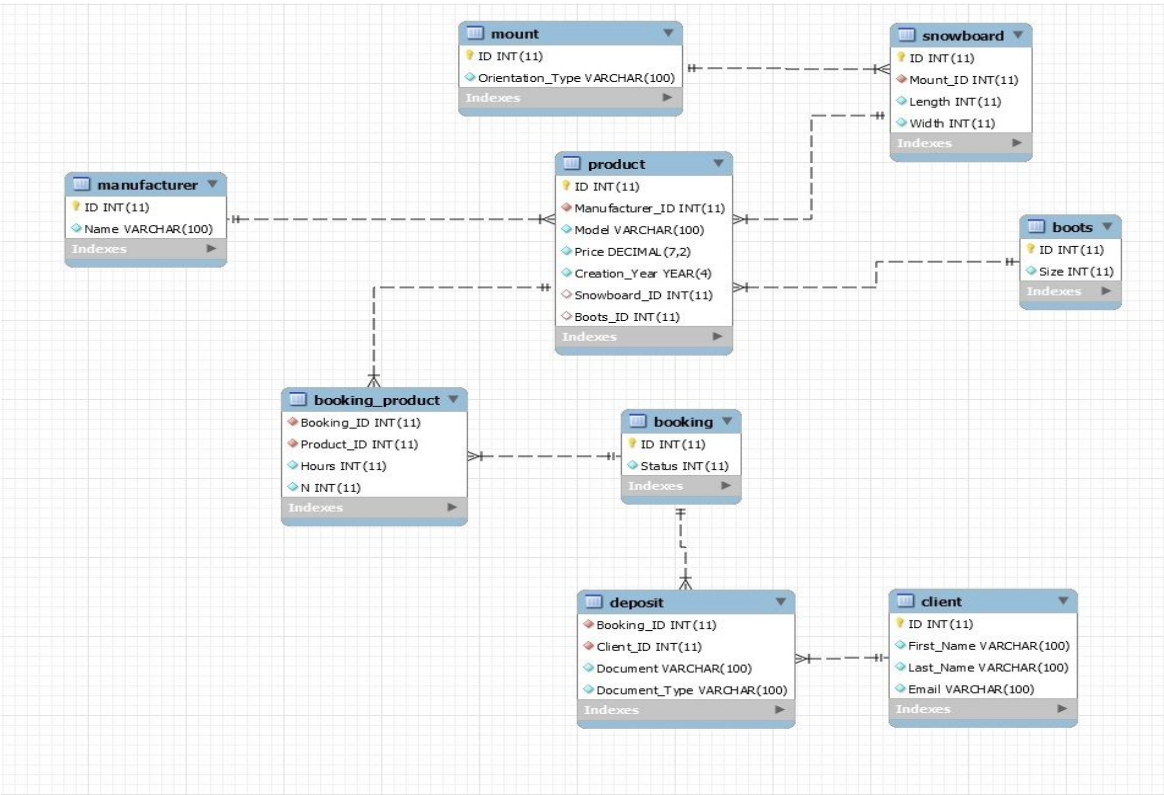
- сноубордические доски - производитель, длина (ростовка), широкая или нет, год выпуска, ориентация креплений на доске (правая нога спереди или сзади)
- ботинки - производитель, размер, год выпуска, модель

Известно, что один и тот же производитель может выпускать как доски, так и обувь.

Клиенты, о который нужно хранить минимальную информацию, включая контактные данные, могут брать в прокат любое количество предметов, оставляя в залог документ, вид и номер которого фиксируются в системе.

Оплата почасовая.

Модель данных



Функциональность

Серверная часть

1. Триггеры

При добавлении товара проверка положительности цены	checkInsertPriceTrigger
При создании заказа проверка того, что он не пуст	
...	
При изменении данных товара проверка положительности цены	checkUpdatePriceTrigger
При изменении данных товара проверка положительности размеров	
При изменении заказа проверка на наличие типа	
...	
Запрет удаления заказов, который еще не закрыты	checkDeleteBookingTrigger
Запрет удаления продукта, заказ на который еще не закрыт	
...	

2. Функции / Процедуры

Добавить клиента	addClient
Изменить информацию об участнике	
...	
Добавить продукт	
Добавить сноуборд в список продуктов	addSnowboard
Добавить ботинки в список продуктов	addBoots

Изменить информацию о продукте	
...	
Подсчитать сумму заказа	countBookingPrice
Подсчитать сумму всех заказов клиента	
...	

3. Представления

Информация про продукты	Product_View
Информация про сноуборды	Snowboards_View
Информация про ботинки	Boots_View
...	
Информация про клиента	
Информация про заказ	
...	
Информация про производителя	
Информация про крепежи	
...	

Клиентская часть

Экранные формы — основные	Экранные формы — дополнительные	Реализация (запрос)	Что здесь можно использовать из серверной части
Клиент			
	Новый участник		addClient
	Изменить данные участника		
	Удалить участника		
	Сортировка	(2) Сортировка клиентов по количеству заказов	
		(3) Сортировка клиентов по общей сумме всех заказов	countBookingPrice
	Информация	(1) Базовая информация про клиента	
Заказ			Booking_View
	Добавить заказ		
	Изменить заказ		
	Удалить заказ		
	Информация	(4) Список заказов с общей суммой	countBookingPrice
		(11) Количество заказов каждого продукта	Booking_View Product_View
		(12) Количество заказов каждого производителя	Booking_View
Товар			Product_View
	Добавить товар		addSnowboard addBoots

	Изменить товар		
	Удалить товар		
	Сортировка	(11) Сортировка товаров по цене	Product_View
	Информация	(5) Список товаров, которые еще не вернули	
		(6) Общая информация про товары	Product_View
		(7) Список товаров, которые брали в аренду	
		(13) Количество товаров по году создания	Product_View
Производитель			
	Добавить производителя		
	Изменить производителя		
	Удалить производителя		
	Сортировка	(12) Количество заказов каждого производителя	Booking_View
Сноуборд			Snowboards_View
	Добавить сноуборд		addSnowboard
	Изменить сноуборд		
	Удалить сноуборд		
	Сортировка	(9) Сортировка сноубордов по цене	Snowboards_View
Ботинки			Boots_View
	Добавить		addBoots

	ботинки		
	Изменить ботинки		
	Удалить ботинки		
	Сортировка	(10) Сортировка ботинок по цене	Boots_View

Скрипты

Серверная часть

Триггеры

```
delimiter //
CREATE TRIGGER checkInsertPriceTrigger BEFORE INSERT ON Product
           FOR EACH ROW
    BEGIN
       IF NEW.Price<0 THEN
          SIGNAL sqlstate '45001' set message_text = "Product.Price must be >= 0";
       END IF;
   END;

call addBoots('F*CK FREEWORLD', -1, 1, 2018, 43);


delimiter //
CREATE TRIGGER checkUpdatePriceTrigger BEFORE UPDATE ON Product
           FOR EACH ROW
    BEGIN
       IF NEW.Price < 0 THEN
          SIGNAL sqlstate '45001' set message_text = "Product.Price must be >= 0";
       END IF;
   END;

update Product set product.Price = -1
          where product.ID = 1


delimiter //
CREATE TRIGGER checkDeleteBookingTrigger BEFORE DELETE ON booking
           FOR EACH ROW
    BEGIN
       IF OLD.Status = 0 THEN
          SIGNAL sqlstate '45001' set message_text = "You can't delete not ended booking";
       END IF;
   END;

delete from booking where booking.ID=2;
```

Хранимые процедуры и функции

```
DELIMITER //
create procedure addClient
        (in first VARCHAR(100),
   in last VARCHAR(100),
   in email VARCHAR(100))
begin
        INSERT INTO Client(First_Name, Last_Name, Email) VALUES (first, last, email);
end;

CALL addClient('Alexandr', 'Ivanov', 'alex.ivanov@yandex.ru');
select * from Client;



DELIMITER //
create procedure addSnowboard
        (
   in Model VARCHAR(100),
   in Price DECIMAL(7,2),
   in Manufacturer_ID INT,
   in Creation_Year INT,
   in Mount_ID INT,
   in Length INT,
   in Width INT
   )
begin
        INSERT INTO Snowboard(Mount_ID, Length, Width) values (Mount_ID, Length, Width);
        INSERT INTO Product(Model, Price, Manufacturer_ID, Creation_Year, Snowboard_ID)
VALUES (Model, Price, Manufacturer_ID, Creation_Year, (SELECT Auto_increment FROM
information_schema.tables WHERE table_name='Snowboard') - 1);
end;

call addSnowboard('FREEK FREEWORLD', 28500, 5, 2018, 2, 150, 35);
select * from product;
```

```
DELIMITER //
create procedure addBoots
        (
   in Model VARCHAR(100),
   in Price DECIMAL(7,2),
   in Manufacturer_ID INT,
   in Creation_Year INT,
   in Size INT
   )
begin
        INSERT INTO Boots(Size) values (Size);
        INSERT INTO Product(Model, Price, Manufacturer_ID, Creation_Year, Boots_ID)
VALUES (Model, Price, Manufacturer_ID, Creation_Year, (SELECT Auto_increment FROM
information_schema.tables WHERE table_name='Boots') - 1);
end;

call addBoots('FREEWORLD LIFE', 13750, 1, 2019, 43);
select * from product;


DELIMITER $$
create function countBookingPrice(id INT) returns DECIMAL(10,2)
deterministic
begin
        return (select sum(product.Price * booking_product.N) from booking
                        inner join booking_product on booking.ID =
booking_product.Booking_ID
                        inner join product on booking_product.Product_ID = product.ID
                                where booking.ID = id);
end;

select countBookingPrice(1);
```

Представления

```
CREATE VIEW Snowboards_View
AS
SELECT snowboard.ID as ID,
                    manufacturer.Name as Manufacturer,
       product.Model as Model,
       snowboard.Length,
       snowboard.Width,
       mount.Orientation_Type,
       product.Price,
       product.Creation_Year
       FROM snowboard
          INNER JOIN Mount on snowboard.Mount_ID = Mount.ID
          INNER JOIN Product on Product.Snowboard_ID = snowboard.ID
    INNER JOIN Manufacturer on Product.Manufacturer_ID = Manufacturer.ID;


CREATE VIEW Boots_View
AS
SELECT boots.ID as ID,
                    manufacturer.Name as Manufacturer,
       product.Model as Model,
       boots.Size,
       product.Price,
       product.Creation_Year
       FROM boots
          INNER JOIN Product on Product.Snowboard_ID = boots.ID
    INNER JOIN Manufacturer on Product.Manufacturer_ID = Manufacturer.ID;
```

```
CREATE VIEW Product_View
AS
select product.ID,
                    'SNOWBOARD' as Product_Type,
                    manufacturer.Name as Manufacturer,
       product.Model,
       product.Price,
       product.Creation_Year from product
          inner join manufacturer on manufacturer.ID = product.Manufacturer_ID
                    where product.Snowboard_ID is not null
union
select product.ID,
                    'BOOTS' as Product_Type,
                    manufacturer.Name as Manufacturer,
       product.Model,
       product.Price,
       product.Creation_Year from product
          inner join manufacturer on manufacturer.ID = product.Manufacturer_ID
                    where product.Boots_ID is not null
          order by Manufacturer;


CREATE VIEW Booking_View
as
select booking.ID,
                    booking.Status,
       countBookingPrice(booking.ID) as BookingPrice,
       client.First_Name,
       client.Last_Name,
       client.Email,
       deposit.Document_Type as DepositDocumentType,
       deposit.Document as DepositDocument
                    from booking
          inner join Deposit on deposit.Booking_ID = booking.ID
    inner join client on client.ID = deposit.Client_ID;
```

Клиентская часть

```
-- (1) Базовая информация про клиента
select * from client;


-- (2) Сортировка клиентов по количеству заказов
select client.ID, client.Email, c.Count from client,
        (select client.ID as ID, count(*) as Count from client
                inner join deposit on deposit.Client_ID = client.ID
                        group by client.ID
         union
    select client.ID as ID, 0 as Count from client
                left join deposit on deposit.Client_ID = client.ID
                        where deposit.Client_ID is null) as c
                        where c.ID = client.ID
        order by c.Count DESC;


-- (3)  Сортировка клиентов по общей сумме всех заказов
select client.ID, client.Email, sum(countBookingPrice(booking.ID)) as BookingSum from client
        inner join deposit on deposit.Client_ID = client.ID
   inner join booking on booking.ID = deposit.Booking_ID
                group by client.ID
union
select client.ID, client.Email, 0 from client
                left join deposit on deposit.Client_ID = client.ID
                        where deposit.Client_ID is null
   order by BookingSum DESC;


-- (4) Список заказов с суммой
select ID, countBookingPrice(ID), Status from booking;


-- (5) Список товаров, которые еще не вернули
select manufacturer.Name, product.Model, product.Price from product
        inner join manufacturer on manufacturer.ID = product.Manufacturer_ID
        inner join booking_product on booking_product.Product_ID = product.ID
   inner join booking on booking.ID = booking_product.Booking_ID
                where booking.Status = 0
                        group by manufacturer.Name, product.Model
                                order by booking.ID;
```

```
-- (6) Общая информация про товары
select * from product_view
	order by product_view.ID;


-- (7) Список товаров, которые брали в аренду
select manufacturer.Name, product.Model, product.Price from product
	inner join manufacturer on manufacturer.ID = product.Manufacturer_ID
	inner join booking_product on booking_product.Product_ID = product.ID
   inner join booking on booking.ID = booking_product.Booking_ID
		where booking.Status = 1
			group by manufacturer.Name, product.Model
				order by booking.ID;


-- (8) Сортировка сноубордов по цене
select * from snowboards_view
	order by Price;


-- (9) Сортировка ботинок по цене
select * from boots_view
	order by Price;


-- (10) Сортировка товаров по цене
select * from product_view
	order by Price;


-- (11) Количество заказов каждого продукта
select pr.ID, pr.Manufacturer, pr.Model, sum(booking_product.N) as Count from product_view
as pr
	inner join booking_product on booking_product.Product_ID = pr.ID
		group by pr.ID
union
select pr.ID, pr.Manufacturer, pr.Model, 0 from product_view as pr
	left join booking_product on booking_product.Product_ID = pr.ID
		where booking_product.Booking_ID is null;


-- (12) Количество заказов каждого производителя
```

```
select manufacturer.ID, manufacturer.Name as Manufacturer, sum(product_count.Count) as
Count  from manufacturer
	inner join
(select pr.Manufacturer as Manufacturer, pr.Model, sum(booking_product.N) as Count from
product_view as pr
inner join booking_product on booking_product.Product_ID = pr.ID
					group by pr.ID
				union
				select pr.Manufacturer as Manufacturer, pr.Model, 0 from
product_view as pr
					left join booking_product on booking_product.Product_ID =
pr.ID
						where booking_product.Booking_ID is null) as
product_count on manufacturer.Name = product_count.Manufacturer
		group by manufacturer.ID
union
select manufacturer.ID, manufacturer.Name AS Manufacturer, 0 from manufacturer
	left join (select pr.Manufacturer as Manufacturer, pr.Model, sum(booking_product.N) as
Count from product_view as pr
				inner join booking_product on booking_product.Product_ID = pr.ID
					group by pr.ID
				union
				select pr.Manufacturer as Manufacturer, pr.Model, 0 from
product_view as pr
					left join booking_product on booking_product.Product_ID =
pr.ID
						where booking_product.Booking_ID is null) as
product_count on manufacturer.Name = product_count.Manufacturer
		where product_count.Manufacturer is null
	order by Manufacturer;


-- (13) Количество товаров по году создания
select product.Creation_Year, count(*) from product
	group by product.Creation_Year
		having product.Creation_Year > 2015;
```

Создание и заполнение базы данных

```
-----------------------------------------------------------
-- Создание базы данных
-----------------------------------------------------------
CREATE SCHEMA rent_db;
```

```
-------------------------------------------------------------
-- Создание таблиц и PK
-------------------------------------------------------------

CREATE TABLE Manufacturer(
   ID                    INTEGER              NOT NULL     AUTO_INCREMENT,
   Name                  VARCHAR(100)         NOT NULL,
CONSTRAINT Manufacturer_PK PRIMARY KEY (ID)
);

CREATE TABLE Mount(
   ID                                         INTEGER      NOT NULL,
        Orientation_Type                VARCHAR(100)       NOT NULL,
CONSTRAINT Mount_PK PRIMARY KEY (ID)
);

CREATE TABLE Snowboard(
   ID                          INTEGER                     NOT NULL     AUTO_INCREMENT,
        Mount_ID               INTEGER       NOT NULL,
   Length                      INTEGER       NOT NULL,
   Width                       INTEGER       NOT NULL,
CONSTRAINT Snowboard_PK PRIMARY KEY (ID)
);

CREATE TABLE Boots(
   ID                    INTEGER              NOT NULL     AUTO_INCREMENT,
        Size             INTEGER    NOT NULL,
CONSTRAINT Boots_PK PRIMARY KEY (ID)
);

CREATE TABLE Product(
   ID                          INTEGER                     NOT NULL
AUTO_INCREMENT,
        Manufacturer_ID INTEGER        NOT NULL,
   Model                       VARCHAR(100)        NOT NULL,
        Price                  DECIMAL(7,2)NOT NULL,
   Creation_Year       YEAR          NOT NULL,
        Snowboard_ID           INTEGER,
   Boots_ID             INTEGER,
CONSTRAINT Product_PK PRIMARY KEY (ID)
);

CREATE TABLE Booking(
```

```
  ID                    INTEGER                 NOT NULL
AUTO_INCREMENT,
  Status                INTEGER     DEFAULT 0       NOT NULL,
CONSTRAINT Booking_PK PRIMARY KEY (ID)
);

CREATE TABLE Booking_Product(
      Booking_ID          INTEGER     NOT NULL,
  Product_ID        INTEGER     NOT NULL,
  Hours                 INTEGER     NOT NULL,
  N                     INTEGER     NOT NULL
);

CREATE TABLE Client(
  ID                            INTEGER           NOT NULL
AUTO_INCREMENT,
      First_Name                VARCHAR(100)      NOT NULL,
      Last_Name                 VARCHAR(100)      NOT NULL,
  Email                         VARCHAR(100)      NOT NULL,
CONSTRAINT Client_PK PRIMARY KEY (ID)
);

CREATE TABLE Deposit(
  Booking_ID        INTEGER     NOT NULL,
  Client_ID     INTEGER     NOT NULL,
  Document      VARCHAR(100)    NOT NULL,
      Document_Type     VARCHAR(100)      NOT NULL
);
```

```
-------------------------------------------------------------
-- Создание FK
-------------------------------------------------------------
ALTER TABLE Snowboard ADD CONSTRAINT FK_Snowboard_Mount
   FOREIGN KEY (Mount_ID)
   REFERENCES Mount(ID)
;

ALTER TABLE Product ADD CONSTRAINT FK_Product_Manufacturer
   FOREIGN KEY (Manufacturer_ID)
   REFERENCES Manufacturer(ID)
;

ALTER TABLE Product ADD CONSTRAINT FK_Product_Snowboard
   FOREIGN KEY (Snowboard_ID)
   REFERENCES Snowboard(ID)
;

ALTER TABLE Product ADD CONSTRAINT FK_Product_Boots
   FOREIGN KEY (Boots_ID)
   REFERENCES Boots(ID)
;

ALTER TABLE Booking_Product ADD CONSTRAINT FK_Booking_Product_Product
   FOREIGN KEY (Product_ID)
   REFERENCES Product(ID)
;

ALTER TABLE Booking_Product ADD CONSTRAINT FK_Booking_Product_Booking
   FOREIGN KEY (Booking_ID)
   REFERENCES Booking(ID)
;

ALTER TABLE Deposit ADD CONSTRAINT FK_Deposit_Booking
   FOREIGN KEY (Booking_ID)
   REFERENCES Booking(ID)
;

ALTER TABLE Deposit ADD CONSTRAINT FK_Deposit_Client
   FOREIGN KEY (Client_ID)
   REFERENCES Client(ID)
;
```

```
-------------------------------------------------------------
-- Заполнение таблиц тестовыми данными
-------------------------------------------------------------

INSERT INTO Manufacturer(ID, Name) VALUES (1, 'ARBOR');
INSERT INTO Manufacturer(ID, Name) VALUES (2, 'BF SNOWBOARDS');
INSERT INTO Manufacturer(ID, Name) VALUES (3, 'BURTON');
INSERT INTO Manufacturer(ID, Name) VALUES (4, 'CAPITA');
INSERT INTO Manufacturer(ID, Name) VALUES (5, 'DC');
INSERT INTO Manufacturer(ID, Name) VALUES (6, 'FLOW');
INSERT INTO Manufacturer(ID, Name) VALUES (7, 'FURBERG');
INSERT INTO Manufacturer(ID, Name) VALUES (8, 'GNU');
INSERT INTO Manufacturer(ID, Name) VALUES (9, 'HEAD');
INSERT INTO Manufacturer(ID, Name) VALUES (10, 'JONES');
INSERT INTO Manufacturer(ID, Name) VALUES (11, 'KORUA SHAPES');
INSERT INTO Manufacturer(ID, Name) VALUES (12, 'LIB TECH');
INSERT INTO Manufacturer(ID, Name) VALUES (13, 'NEVER SUMMER');
INSERT INTO Manufacturer(ID, Name) VALUES (14, 'ROME');
INSERT INTO Manufacturer(ID, Name) VALUES (15, 'ROXY');
INSERT INTO Manufacturer(ID, Name) VALUES (16, 'SALOMON');
INSERT INTO Manufacturer(ID, Name) VALUES (17, 'SIGNAL');
INSERT INTO Manufacturer(ID, Name) VALUES (18, 'YES');

INSERT INTO Mount(ID, Orientation_Type) VALUES (1, 'RIGHT');
INSERT INTO Mount(ID, Orientation_Type) VALUES (2, 'LEFT');


INSERT INTO Boots(ID, Size) VALUES (1, 42);
INSERT INTO Boots(ID, Size) VALUES (2, 43);
INSERT INTO Boots(ID, Size) VALUES (3, 36);
INSERT INTO Boots(ID, Size) VALUES (4, 45);
INSERT INTO Boots(ID, Size) VALUES (5, 45);
INSERT INTO Boots(ID, Size) VALUES (6, 44);
INSERT INTO Boots(ID, Size) VALUES (7, 32);
INSERT INTO Boots(ID, Size) VALUES (8, 32);
INSERT INTO Boots(ID, Size) VALUES (9, 35);
INSERT INTO Boots(ID, Size) VALUES (10, 42);
INSERT INTO Boots(ID, Size) VALUES (11, 36);
INSERT INTO Boots(ID, Size) VALUES (12, 46);
INSERT INTO Boots(ID, Size) VALUES (13, 34);
INSERT INTO Boots(ID, Size) VALUES (14, 42);
INSERT INTO Boots(ID, Size) VALUES (15, 38);
INSERT INTO Boots(ID, Size) VALUES (16, 41);
```

```
INSERT INTO Boots(ID, Size) VALUES (17, 39);
INSERT INTO Boots(ID, Size) VALUES (18, 40);
INSERT INTO Boots(ID, Size) VALUES (19, 39);
INSERT INTO Boots(ID, Size) VALUES (20, 33);

INSERT INTO Snowboard(ID, Mount_ID, Length, Width) VALUES (1, 1, 240, 20);
INSERT INTO Snowboard(ID, Mount_ID, Length, Width) VALUES (2, 1, 210, 50);
INSERT INTO Snowboard(ID, Mount_ID, Length, Width) VALUES (3, 1, 290, 30);
INSERT INTO Snowboard(ID, Mount_ID, Length, Width) VALUES (4, 2, 170, 30);
INSERT INTO Snowboard(ID, Mount_ID, Length, Width) VALUES (5, 2, 290, 10);
INSERT INTO Snowboard(ID, Mount_ID, Length, Width) VALUES (6, 1, 150, 40);
INSERT INTO Snowboard(ID, Mount_ID, Length, Width) VALUES (7, 1, 200, 40);
INSERT INTO Snowboard(ID, Mount_ID, Length, Width) VALUES (8, 2, 160, 40);
INSERT INTO Snowboard(ID, Mount_ID, Length, Width) VALUES (9, 2, 180, 10);
INSERT INTO Snowboard(ID, Mount_ID, Length, Width) VALUES (10, 2, 150, 50);
INSERT INTO Snowboard(ID, Mount_ID, Length, Width) VALUES (11, 2, 230, 50);
INSERT INTO Snowboard(ID, Mount_ID, Length, Width) VALUES (12, 2, 200, 50);
INSERT INTO Snowboard(ID, Mount_ID, Length, Width) VALUES (13, 2, 260, 50);
INSERT INTO Snowboard(ID, Mount_ID, Length, Width) VALUES (14, 2, 240, 10);
INSERT INTO Snowboard(ID, Mount_ID, Length, Width) VALUES (15, 1, 160, 40);
INSERT INTO Snowboard(ID, Mount_ID, Length, Width) VALUES (16, 1, 260, 20);
INSERT INTO Snowboard(ID, Mount_ID, Length, Width) VALUES (17, 1, 210, 50);
INSERT INTO Snowboard(ID, Mount_ID, Length, Width) VALUES (18, 2, 200, 20);
INSERT INTO Snowboard(ID, Mount_ID, Length, Width) VALUES (19, 2, 180, 40);
INSERT INTO Snowboard(ID, Mount_ID, Length, Width) VALUES (20, 2, 160, 50);

INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Boots_ID)
        VALUES (1, 'ION', 15000, 16, 2015, 1);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Boots_ID)
        VALUES (2, 'MINT', 44000, 7, 2019, 2);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Boots_ID)
        VALUES (3, 'PHOTON STEP ON', 36200, 8, 2019, 3);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Boots_ID)
        VALUES (4, 'MINT', 31000, 5, 2013, 4);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Boots_ID)
        VALUES (5, 'MINT', 17100, 9, 2014, 5);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Boots_ID)
        VALUES (6, 'ION', 49400, 8, 2010, 6);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Boots_ID)
        VALUES (7, 'ION', 46200, 16, 2017, 7);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Boots_ID) VALUES
(8, 'MINT', 18100, 15, 2012, 8);
```

```
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Boots_ID) VALUES (9, 'IMPERIAL', 45300, 12, 2012, 9);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Boots_ID) VALUES (10, 'ION STEP ON', 32600, 3, 2014, 10);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Boots_ID) VALUES (11, 'ION', 23700, 5, 2017, 11);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Boots_ID) VALUES (12, 'ION', 38000, 7, 2018, 12);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Boots_ID) VALUES (13, 'ION STEP ON', 17200, 12, 2017, 13);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Boots_ID) VALUES (14, 'RULER BOA', 30200, 8, 2019, 14);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Boots_ID) VALUES (15, 'ION', 28900, 7, 2016, 15);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Boots_ID) VALUES (16, 'PHOTON STEP ON', 49600, 13, 2018, 16);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Boots_ID) VALUES (17, 'ION', 43100, 11, 2012, 17);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Boots_ID) VALUES (18, 'RULER BOA', 29500, 3, 2019, 18);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Boots_ID) VALUES (19, 'IMPERIAL', 11100, 5, 2015, 19);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Boots_ID) VALUES (20, 'ION', 48400, 15, 2018, 20);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Snowboard_ID) VALUES (21, 'CUSTOM X', 46400, 2, 2013, 1);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Snowboard_ID) VALUES (22, 'SK8 BANANA BTX', 27300, 6, 2013, 2);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Snowboard_ID) VALUES (23, 'DIY THROWBACK', 53400, 1, 2017, 3);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Snowboard_ID) VALUES (24, 'PROCESS SMALLS', 37700, 13, 2019, 4);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Snowboard_ID) VALUES (25, 'CUSTOM X', 74700, 8, 2010, 5);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Snowboard_ID) VALUES (26, 'CAFE RACER', 32500, 18, 2018, 6);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Snowboard_ID) VALUES (27, 'PROCESS SMALLS', 54200, 10, 2014, 7);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Snowboard_ID) VALUES (28, 'FRONTIER', 35700, 17, 2016, 8);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Snowboard_ID) VALUES (29, 'MECHANIC', 24000, 10, 2016, 9);
```

```
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Snowboard_ID)
VALUES (30, 'FOCUS M SNBD', 51800, 18, 2013, 10);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Snowboard_ID)
VALUES (31, 'SK8 BANANA BTX', 66800, 12, 2014, 11);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Snowboard_ID)
VALUES (32, 'CAFE RACER', 71600, 3, 2016, 12);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Snowboard_ID)
VALUES (33, 'THE NAVIGATOR', 67400, 11, 2016, 13);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Snowboard_ID)
VALUES (34, 'CHOPPER', 46000, 17, 2013, 14);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Snowboard_ID)
VALUES (35, 'CAFE RACER', 61800, 11, 2012, 15);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Snowboard_ID)
VALUES (36, 'CAFE RACER', 38400, 11, 2019, 16);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Snowboard_ID)
VALUES (37, 'CAFE RACER', 32200, 9, 2012, 17);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Snowboard_ID)
VALUES (38, 'CHOPPER', 74100, 9, 2013, 18);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Snowboard_ID)
VALUES (39, 'PROCESS SMALLS', 59200, 7, 2016, 19);
INSERT INTO Product(ID, Model, Price, Manufacturer_ID, Creation_Year, Snowboard_ID)
VALUES (40, 'PROCESS SMALLS', 74000, 7, 2017, 20);


INSERT INTO Booking(ID, Status) VALUES (1, 1);
INSERT INTO Booking(ID, Status) VALUES (2, 0);
INSERT INTO Booking(ID, Status) VALUES (3, 0);
INSERT INTO Booking(ID, Status) VALUES (4, 0);
INSERT INTO Booking(ID, Status) VALUES (5, 1);
INSERT INTO Booking(ID, Status) VALUES (6, 0);
INSERT INTO Booking(ID, Status) VALUES (7, 1);
INSERT INTO Booking(ID, Status) VALUES (8, 0);
INSERT INTO Booking(ID, Status) VALUES (9, 1);
INSERT INTO Booking(ID, Status) VALUES (10, 1);
INSERT INTO Booking(ID, Status) VALUES (11, 0);
INSERT INTO Booking(ID, Status) VALUES (12, 1);
INSERT INTO Booking(ID, Status) VALUES (13, 1);
INSERT INTO Booking(ID, Status) VALUES (14, 1);
INSERT INTO Booking(ID, Status) VALUES (15, 1);
INSERT INTO Booking(ID, Status) VALUES (16, 1);
INSERT INTO Booking(ID, Status) VALUES (17, 0);
INSERT INTO Booking(ID, Status) VALUES (18, 0);
INSERT INTO Booking(ID, Status) VALUES (19, 0);
```

INSERT INTO Booking(ID, Status) VALUES (20, 0);

INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (9, 40, 11, 1);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (19, 33, 16, 4);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (2, 11, 9, 3);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (7, 14, 9, 4);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (10, 19, 6, 1);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (3, 17, 22, 2);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (11, 38, 22, 5);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (18, 11, 11, 3);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (4, 3, 22, 1);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (20, 39, 14, 1);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (1, 39, 23, 3);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (9, 6, 16, 3);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (7, 21, 6, 3);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (19, 18, 14, 4);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (3, 6, 23, 2);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (2, 24, 18, 1);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (2, 40, 16, 2);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (6, 2, 12, 1);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (18, 22, 14, 5);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (3, 27, 14, 3);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (5, 19, 13, 1);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (14, 14, 9, 1);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (4, 22, 23, 2);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (10, 10, 22, 5);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (16, 11, 10, 4);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (14, 8, 1, 3);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (19, 10, 17, 2);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (14, 13, 5, 1);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (5, 35, 23, 4);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (4, 3, 7, 3);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (17, 24, 4, 2);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (8, 1, 2, 1);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (13, 10, 12, 3);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (12, 27, 11, 5);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (12, 22, 15, 4);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (10, 23, 8, 5);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (10, 3, 4, 4);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (10, 17, 18, 1);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (5, 38, 17, 3);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (15, 30, 7, 3);

```
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (7, 17, 17, 3);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (15, 32, 23, 4);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (4, 1, 18, 2);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (5, 4, 21, 5);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (14, 6, 9, 5);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (8, 15, 7, 5);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (7, 21, 14, 4);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (6, 9, 16, 5);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (1, 16, 21, 4);
INSERT INTO Booking_Product(Booking_ID, Product_ID, Hours, N) VALUES (19, 36, 6, 4);


INSERT INTO Client(ID, First_Name, Last_Name, Email) VALUES (1, 'Vyacheslav', 'Alikin',
'vyacheslav.alikin@yahoo.ru');
INSERT INTO Client(ID, First_Name, Last_Name, Email) VALUES (2, 'Bojan', 'Golubev',
'bojan.golubev@yandex.ru');
INSERT INTO Client(ID, First_Name, Last_Name, Email) VALUES (3, 'Pavel', 'Alikin',
'pavel.alikin@yandex.ru');
INSERT INTO Client(ID, First_Name, Last_Name, Email) VALUES (4, 'Bojan', 'Aliyev',
'bojan.aliyev@gmail.com');
INSERT INTO Client(ID, First_Name, Last_Name, Email) VALUES (5, 'Aleksandr', 'Aliyev',
'aleksandr.aliyev@gmail.com');
INSERT INTO Client(ID, First_Name, Last_Name, Email) VALUES (6, 'Olivier', 'Chernov',
'olivier.chernov@yahoo.ru');
INSERT INTO Client(ID, First_Name, Last_Name, Email) VALUES (7, 'Bojan', 'Yemelyanov',
'bojan.yemelyanov@mail.ru');
INSERT INTO Client(ID, First_Name, Last_Name, Email) VALUES (8, 'Vyacheslav', 'Sukhov',
'vyacheslav.sukhov@mail.ru');
INSERT INTO Client(ID, First_Name, Last_Name, Email) VALUES (9, 'Denis', 'Sysuyev',
'denis.sysuyev@yandex.ru');
INSERT INTO Client(ID, First_Name, Last_Name, Email) VALUES (10, 'Turgay', 'Shafinsky',
'turgay.shafinsky@yandex.ru');
INSERT INTO Client(ID, First_Name, Last_Name, Email) VALUES (11, 'Daniil', 'Kozlov',
'daniil.kozlov@yandex.ru');
INSERT INTO Client(ID, First_Name, Last_Name, Email) VALUES (12, 'Yury', 'Nikitin',
'yury.nikitin@yahoo.ru');
INSERT INTO Client(ID, First_Name, Last_Name, Email) VALUES (13, 'Dmitri', 'Tabidze',
'dmitri.tabidze@gmail.com');
INSERT INTO Client(ID, First_Name, Last_Name, Email) VALUES (14, 'Kirill', 'Golubev',
'kirill.golubev@yahoo.ru');
INSERT INTO Client(ID, First_Name, Last_Name, Email) VALUES (15, 'Nikolai', 'Krugovoy',
'nikolai.krugovoy@yandex.ru');
```

```
INSERT INTO Client(ID, First_Name, Last_Name, Email) VALUES (16, 'Denis', 'Sukhov',
'denis.sukhov@mail.ru');
INSERT INTO Client(ID, First_Name, Last_Name, Email) VALUES (17, 'Aleksandr', 'Carp',
'aleksandr.carp@yandex.ru');
INSERT INTO Client(ID, First_Name, Last_Name, Email) VALUES (18, 'Daniil', 'Krotov',
'daniil.krotov@yandex.ru');
INSERT INTO Client(ID, First_Name, Last_Name, Email) VALUES (19, 'Aleksei', 'Shafinsky',
'aleksei.shafinsky@mail.ru');
INSERT INTO Client(ID, First_Name, Last_Name, Email) VALUES (20, 'Kirill', 'Krugovoy',
'kirill.krugovoy@mail.ru');


INSERT INTO Deposit(Booking_ID, Client_ID, Document, Document_Type) VALUES (1, 17,
'4007 688408', 'Student ID');
INSERT INTO Deposit(Booking_ID, Client_ID, Document, Document_Type) VALUES (2, 16,
'4009 61820', 'Driver License');
INSERT INTO Deposit(Booking_ID, Client_ID, Document, Document_Type) VALUES (3, 7,
'4008 595561', 'Student ID');
INSERT INTO Deposit(Booking_ID, Client_ID, Document, Document_Type) VALUES (4, 1,
'4000 650069', 'International Passport');
INSERT INTO Deposit(Booking_ID, Client_ID, Document, Document_Type) VALUES (5, 3,
'4010 769005', 'Student ID');
INSERT INTO Deposit(Booking_ID, Client_ID, Document, Document_Type) VALUES (6, 12,
'4013 588286', 'Passport');
INSERT INTO Deposit(Booking_ID, Client_ID, Document, Document_Type) VALUES (7, 15,
'4010 403512', 'International Passport');
INSERT INTO Deposit(Booking_ID, Client_ID, Document, Document_Type) VALUES (8, 12,
'4000 945197', 'Student ID');
INSERT INTO Deposit(Booking_ID, Client_ID, Document, Document_Type) VALUES (9, 12,
'4016 741244', 'Student ID');
INSERT INTO Deposit(Booking_ID, Client_ID, Document, Document_Type) VALUES (10, 13,
'4007 150563', 'Passport');
INSERT INTO Deposit(Booking_ID, Client_ID, Document, Document_Type) VALUES (11, 18,
'4016 505584', 'International Passport');
INSERT INTO Deposit(Booking_ID, Client_ID, Document, Document_Type) VALUES (12, 10,
'4017 932072', 'International Passport');
INSERT INTO Deposit(Booking_ID, Client_ID, Document, Document_Type) VALUES (13, 3,
'4017 751252', 'Student ID');
INSERT INTO Deposit(Booking_ID, Client_ID, Document, Document_Type) VALUES (14, 17,
'4018 925583', 'Student ID');
INSERT INTO Deposit(Booking_ID, Client_ID, Document, Document_Type) VALUES (15, 2,
'4001 540428', 'Driver License');
```

```
INSERT INTO Deposit(Booking_ID, Client_ID, Document, Document_Type) VALUES (16, 8, '4009 54630', 'Driver License');
INSERT INTO Deposit(Booking_ID, Client_ID, Document, Document_Type) VALUES (17, 11, '4017 459459', 'International Passport');
INSERT INTO Deposit(Booking_ID, Client_ID, Document, Document_Type) VALUES (18, 16, '4016 893130', 'Passport');
INSERT INTO Deposit(Booking_ID, Client_ID, Document, Document_Type) VALUES (19, 6, '4008 489965', 'International Passport');
INSERT INTO Deposit(Booking_ID, Client_ID, Document, Document_Type) VALUES (20, 19, '4010 822312', 'Student ID');
```

```
-------------------------------------------------------------
-- Вывод таблиц
-------------------------------------------------------------
select * from Client;
select * from booking_product;
select * from Booking;
select * from Deposit;
select * from Product;
select * from Boots;
select * from Snowboard;
select * from Mount;
select * from Manufacturer;
```

```
-------------------------------------------------------------
-- Удаление таблиц
-------------------------------------------------------------

DROP TABLE Deposit;
DROP TABLE Client;
DROP TABLE Booking_Product;
DROP TABLE Product;
DROP TABLE Booking;
DROP TABLE Boots;
DROP TABLE Snowboard;
DROP TABLE Mount;
DROP TABLE Manufacturer;
```