Б05-811

Сеть баров

Проект базы данных

Айгожиев Бахтияр

Логическая модель:

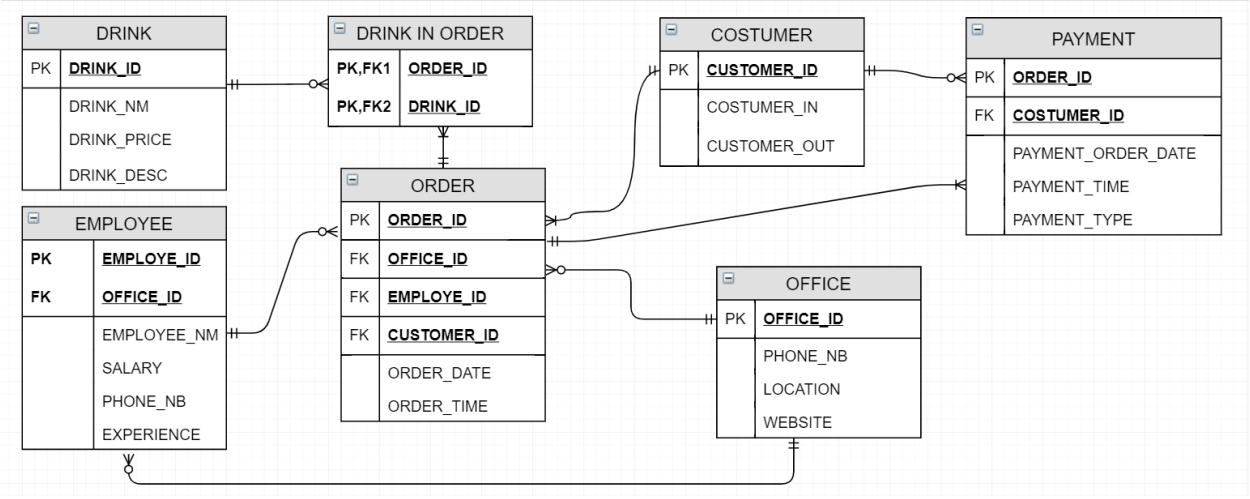

Схема бара:

Схема «Bar» состоит из 7 сущностей

Drink – “Меню” бара, содержит поля с данными о напитке: номер, название, описание и цену

Drink in order – содержит данные о напитках в заказе: номер напитка и номер заказа

Order – содержит данные о заказе: номер заказа, номер филиала, номер сотрудника, номер клиента, дата оформления заказа и время оформления заказа

Costumer – содержит данные о клиенте: номер клиента, время входа клиента и время выхода клиента

Payment – содержит данные о оплате заказа: номер заказа, номер клиента, дата оформления оплаты, время оформления оплаты и тип оплаты

Office – содержит информацию о филиале бара: номер филиала, номер телефона, местоположение и веб-сайт

Employee – содержит информацию о сотруднике бара: номер, номер филиала, имя сотрудника, зарплату, мобильный телефон, опыт

Функции:

Order_price (*id_ord* int) – по номеру заказа определяет его стоимость

Most_popular_drinks (day date) – По дате возвращает самый покупаемый напиток

Триггеры:

Emp_update – при вставке данных в таблицу payment увеличивает опыт сотрудника в таблице employee

Роли и права:

Director – может смотреть таблицу Office, изменять таблицу Employee

Manager – может смотреть таблицу Employee, изменять таблицы Order и Drink_in_order

Barman – может изменять таблицу Drink

Код:

```
CREATE SCHEMA BAR;

CREATE TABLE BAR.Costumer (
   COSTUMER_ID int NOT NULL,
   COSTUMER_IN TIMESTAMPTZ,
   COSTUMER_OUT TIMESTAMPTZ,
   CONSTRAINT Costumer_pk PRIMARY KEY (COSTUMER_ID)
);

CREATE TABLE BAR.DRINK (
   DRINK_ID int NOT NULL,
   DRINK_NM varchar(20) NOT NULL,
   DRINK_PRICE int NOT NULL,
   DRINK_DESC TEXT NOT NULL,
   CONSTRAINT DRINK_pk PRIMARY KEY (DRINK_ID),
   CONSTRAINT CHK_DRINK CHECK (DRINK_ID > 0 and DRINK_PRICE > 0)
);


CREATE TABLE BAR.OFFICE (
   OFFICE_ID int NOT NULL,
   LOCATION varchar(100) NOT NULL,
   PHONE_NB varchar(10) NOT NULL,
   WEBSITE varchar(25) NOT NULL,
   CONSTRAINT OFFICE_PK PRIMARY KEY (OFFICE_ID),
   CONSTRAINT CHK_OFFICE CHECK (OFFICE_ID > 0 and PHONE_NB not like '%[^0-9]%')
);


CREATE TABLE BAR.EMPLOYEE (
   EMPLOYEE_ID int NOT NULL,
```

```
   OFFICE_ID int NOT NULL,
   EMPLOYEE_NM varchar(25) NOT NULL,
   SALARY int NOT NULL,
   PHONE_NB varchar(10) NOT NULL,
   EXPERIENCE int NOT NULL,
   CONSTRAINT EMPLOYEE_PK PRIMARY KEY (EMPLOYEE_ID),
   CONSTRAINT OFFICE_EMPLOYEE FOREIGN KEY (OFFICE_ID)
   REFERENCES BAR.OFFICE (OFFICE_ID),
   CONSTRAINT CHK_EMPLOYEE CHECK (EMPLOYEE_ID > 0 and salary > 10000 and PHONE_NB not like
'%[^0-9]%'
                              and experience >= 0)
);



CREATE TABLE BAR.ORDER (
   ORDER_ID int NOT NULL,
   OFFICE_ID int NOT NULL,
   COSTUMER_ID int NOT NULL,
   EMPLOYEE_ID int NOT NULL,
   ORDER_DATE DATE NOT NULL,
   ORDER_TIME TIME NOT NULL,
   CONSTRAINT ORDER_PK PRIMARY KEY (ORDER_ID),
   CONSTRAINT ORDERER_Costumer FOREIGN KEY (COSTUMER_ID)
   REFERENCES BAR. Costumer (COSTUMER_ID),
   CONSTRAINT ORDER_OFFICE FOREIGN KEY (OFFICE_ID)
   REFERENCES BAR.OFFICE (OFFICE_ID),
   CONSTRAINT ORDER_EMPLOYEE FOREIGN KEY (EMPLOYEE_ID)
   REFERENCES BAR.EMPLOYEE (EMPLOYEE_ID),
   CONSTRAINT CHK_ORDER CHECK (ORDER_ID > 0)
);



CREATE TABLE BAR.DRINK_IN_ORDER (
```

```
    DRINK_ID int NOT NULL,
    ORDER_ID int NOT NULL,
    CONSTRAINT DRINK_IN_ORDER_DRINK FOREIGN KEY (DRINK_ID)
    REFERENCES BAR.DRINK (DRINK_ID),
    CONSTRAINT DRINK_IN_ORDER_ORDER FOREIGN KEY (ORDER_ID)
    REFERENCES BAR.ORDER (ORDER_ID)
);

CREATE TABLE BAR.PAYMENT (
    ORDER_ID int NOT NULL,
    COSTUMER_ID int NOT NULL,
    PAYMENT_ORDER_DATE DATE NOT NULL,
    PAYMENT_time time NOT NULL,
    PAYMENT_TYPE varchar(10) NOT NULL,
    CONSTRAINT PAYMENT_pk PRIMARY KEY (ORDER_ID, COSTUMER_ID),
    CONSTRAINT PAYMENT_Costumer FOREIGN KEY (COSTUMER_ID)
    REFERENCES BAR.Costumer (COSTUMER_ID),
    CONSTRAINT PAYMENT_ORDER FOREIGN KEY (ORDER_ID)
    REFERENCES BAR.ORDER (ORDER_ID)
);

select *
FROM BAR.DRINK
;

INSERT INTO BAR.OFFICE (OFFICE_ID, LOCATION, PHONE_NB, WEBSITE) VALUES (1, 'Pervomayskaya, 32k2', '9851234567', 'www.BAR1.ru');
INSERT INTO BAR.OFFICE (OFFICE_ID, LOCATION, PHONE_NB, WEBSITE) VALUES (2, 'Institutsky per., 9', '9876543210', 'www.BAR2.ru');
INSERT INTO BAR.OFFICE (OFFICE_ID, LOCATION, PHONE_NB, WEBSITE) VALUES (3, 'Pervomayskaya, 42', '9857654321', 'www.BAR3ru');
INSERT INTO BAR.OFFICE (OFFICE_ID, LOCATION, PHONE_NB, WEBSITE) VALUES (4, 'Sovetskaya, 9', '9996547813', 'www.BAR4.ru');
```

```
INSERT INTO BAR.OFFICE (OFFICE_ID, LOCATION, PHONE_NB, WEBSITE) VALUES (5, 'Pavlova, 6',
'9884537653', 'www.BAR5.ru');


INSERT INTO BAR.DRINK (DRINK_ID, DRINK_NM, DRINK_PRICE, DRINK_DESC) VALUES (1, 'WHISKY',
'1000', 'Usual');

INSERT INTO BAR.DRINK (DRINK_ID, DRINK_NM, DRINK_PRICE, DRINK_DESC) VALUES (2, 'VODKA', '500',
'USUAL');

INSERT INTO BAR.DRINK (DRINK_ID, DRINK_NM, DRINK_PRICE, DRINK_DESC) VALUES (3, 'Martini', '300',
'Light');

INSERT INTO BAR.DRINK (DRINK_ID, DRINK_NM, DRINK_PRICE, DRINK_DESC) VALUES (4, 'White russian',
'800', 'Usual coctail');

INSERT INTO BAR.DRINK (DRINK_ID, DRINK_NM, DRINK_PRICE, DRINK_DESC) VALUES (5, 'Submarine',
'1500', 'Usual');

INSERT INTO BAR.DRINK (DRINK_ID, DRINK_NM, DRINK_PRICE, DRINK_DESC) VALUES (6, 'Jin', '500',
'Spicy');

INSERT INTO BAR.DRINK (DRINK_ID, DRINK_NM, DRINK_PRICE, DRINK_DESC) VALUES (7, 'Wine', '1000',
'Light');


INSERT INTO BAR.Costumer (COSTUMER_ID) VALUES (1);

INSERT INTO BAR.Costumer (COSTUMER_ID) VALUES (2);

INSERT INTO BAR.Costumer (COSTUMER_ID) VALUES (3);

INSERT INTO BAR.Costumer (COSTUMER_ID) VALUES (4);

INSERT INTO BAR.Costumer (COSTUMER_ID) VALUES (5);


INSERT INTO BAR.EMPLOYEE (EMPLOYEE_ID, OFFICE_ID, EMPLOYEE_NM, salary, PHONE_NB,
experience) VALUES (1, 1, 'Danil Garkin', 70000, '9768765434', 3);

INSERT INTO BAR.EMPLOYEE (EMPLOYEE_ID, OFFICE_ID, EMPLOYEE_NM, salary, PHONE_NB,
experience) VALUES (2, 1, 'Mikhail Katkov', 30000, '9768098434', 0);

INSERT INTO BAR.EMPLOYEE (EMPLOYEE_ID, OFFICE_ID, EMPLOYEE_NM, salary, PHONE_NB,
experience) VALUES (3, 2, 'Anton Kulyamin', 15000, '9723454734', 0);

INSERT INTO BAR.EMPLOYEE (EMPLOYEE_ID, OFFICE_ID, EMPLOYEE_NM, salary, PHONE_NB,
experience) VALUES (4, 3, 'Sergey Bogdanov', 25000, '9768724111', 0);

INSERT INTO BAR.EMPLOYEE (EMPLOYEE_ID, OFFICE_ID, EMPLOYEE_NM, salary, PHONE_NB,
experience) VALUES (5, 2, 'Danil Pismensky', 100000, '9118098434', 8);

INSERT INTO BAR.EMPLOYEE (EMPLOYEE_ID, OFFICE_ID, EMPLOYEE_NM, salary, PHONE_NB,
experience) VALUES (6, 3, 'Evgeny Nepryahin', 35000, '9674800934', 1);

INSERT INTO BAR.EMPLOYEE (EMPLOYEE_ID, OFFICE_ID, EMPLOYEE_NM, salary, PHONE_NB,
experience) VALUES (7, 4, 'Eduard Nikolaenko', 150000, '9671488228', 5);
```

```
INSERT INTO BAR.EMPLOYEE (EMPLOYEE_ID, OFFICE_ID, EMPLOYEE_NM, salary, PHONE_NB, experience) VALUES (8, 4, 'Darya Mitskaya', 50000, '9677878258', 3);

INSERT INTO BAR.EMPLOYEE (EMPLOYEE_ID, OFFICE_ID, EMPLOYEE_NM, salary, PHONE_NB, experience) VALUES (9, 5, 'Sofiya Samohina', 15000, '9622768168', 5);

INSERT INTO BAR.EMPLOYEE (EMPLOYEE_ID, OFFICE_ID, EMPLOYEE_NM, salary, PHONE_NB, experience) VALUES (10, 5, 'Alexandr Kulinich', 30000, '9683569978', 0);


INSERT INTO BAR.ORDER (ORDER_ID, OFFICE_ID, COSTUMER_ID, EMPLOYEE_ID, ORDER_DATE, ORDER_TIME) VALUES (1, 1, 2, 2, '2019-04-20', '19:34:04');

INSERT INTO BAR.ORDER (ORDER_ID, OFFICE_ID, COSTUMER_ID, EMPLOYEE_ID, ORDER_DATE, ORDER_TIME) VALUES (2, 5, 4, 10, '2019-04-20', '22:31:25');

INSERT INTO BAR.ORDER (ORDER_ID, OFFICE_ID, COSTUMER_ID, EMPLOYEE_ID, ORDER_DATE, ORDER_TIME) VALUES (3, 4, 1, 8, '2019-04-20', '22:31:25');


INSERT INTO BAR.DRINK_IN_ORDER (DRINK_ID, ORDER_ID) VALUES (1, 1);

INSERT INTO BAR.DRINK_IN_ORDER (DRINK_ID, ORDER_ID) VALUES (7, 1);

INSERT INTO BAR.DRINK_IN_ORDER (DRINK_ID, ORDER_ID) VALUES (7, 1);

INSERT INTO BAR.DRINK_IN_ORDER (DRINK_ID, ORDER_ID) VALUES (4, 1);

INSERT INTO BAR.DRINK_IN_ORDER (DRINK_ID, ORDER_ID) VALUES (2, 2);

INSERT INTO BAR.DRINK_IN_ORDER (DRINK_ID, ORDER_ID) VALUES (6, 2);

INSERT INTO BAR.DRINK_IN_ORDER (DRINK_ID, ORDER_ID) VALUES (1, 1);

INSERT INTO BAR.DRINK_IN_ORDER (DRINK_ID, ORDER_ID) VALUES (7, 1);

INSERT INTO BAR.DRINK_IN_ORDER (DRINK_ID, ORDER_ID) VALUES (3, 3);

INSERT INTO BAR.DRINK_IN_ORDER (DRINK_ID, ORDER_ID) VALUES (5, 3);

INSERT INTO BAR.DRINK_IN_ORDER (DRINK_ID, ORDER_ID) VALUES (4, 3);


INSERT INTO BAR.PAYMENT (ORDER_ID, COSTUMER_ID, PAYMENT_ORDER_DATE, PAYMENT_time, PAYMENT_TYPE) VALUES (1, 2, '2019-04-20', '22:44:09', 'Card');

INSERT INTO BAR.PAYMENT (ORDER_ID, COSTUMER_ID, PAYMENT_ORDER_DATE, PAYMENT_time, PAYMENT_TYPE) VALUES (2, 4, '2019-04-21', '00:28:42', 'Card');

INSERT INTO BAR.PAYMENT (ORDER_ID, COSTUMER_ID, PAYMENT_ORDER_DATE, PAYMENT_time, PAYMENT_TYPE) VALUES (3, 1, '2019-04-21', '00:54:39', 'Card');


--Average salary

SELECT AVG(SALARY)
```

```
FROM bar.EMPLOYEE
;

--Costumers with their order's price
SELECT bar.costumer.costumer_id, all_price
FROM bar.Costumer INNER JOIN bar.PAYMENT on Costumer.COSTUMER_ID = PAYMENT.COSTUMER_ID
INNER JOIN (
   SELECT bar.ORDER.ORDER_ID, SUM(DRINK.drink_price) as all_price
   FROM bar.ORDER
      INNER JOIN bar.DRINK_IN_ORDER on bar.ORDER.ORDER_ID = DRINK_IN_ORDER.ORDER_ID
      INNER JOIN bar.DRINK on DRINK_IN_ORDER.DRINK_ID = bar.DRINK.DRINK_ID
   GROUP BY bar.ORDER.ORDER_ID
) AS ORDERers on ORDERers.ORDER_ID = PAYMENT.ORDER_ID;

--Max salary
SELECT EMPLOYEE_NM, salary
FROM bar.EMPLOYEE
WHERE bar.EMPLOYEE.salary in (
 SELECT MAX(salary)
 FROM bar.EMPLOYEE
   );


CREATE VIEW Sum_of_Costumers_ORDER AS (
   WITH Sum_of_ORDER AS (
      SELECT bar.ORDER.ORDER_ID as id_ORDER, SUM(bar.DRINK.drink_price) as s
      FROM bar.ORDER
            INNER JOIN bar.DRINK_IN_ORDER on bar.ORDER.ORDER_ID = DRINK_IN_ORDER.ORDER_ID
            INNER JOIN bar.DRINK on DRINK_IN_ORDER.DRINK_ID = bar.DRINK.DRINK_ID
      GROUP BY bar.ORDER.ORDER_ID
      )
      SELECT bar.Costumer.Costumer_id, Sum_of_ORDER.s
```

```
    FROM bar.Costumer
      INNER JOIN bar.ORDER ON Costumer.COSTUMER_ID = bar.ORDER.COSTUMER_ID
      INNER JOIN Sum_of_ORDER ON Sum_of_ORDER.id_ORDER = bar.ORDER.ORDER_ID
);

CREATE VIEW average_salary_by_office AS (
 SELECT office_id, bar.EMPLOYEE.EMPLOYEE_NM AS Name, salary, AVG(salary) OVER (PARTITION BY office_id) as Average_salary
 FROM bar.EMPLOYEE
);

select *
from average_salary_by_office
order by average_salary_by_office.office_id;

CREATE OR REPLACE FUNCTION Order_price (id_ord int)
RETURNS INT
AS $BODY$
BEGIN
IF (id_ord NOT IN (SELECT ORDER_ID FROM BAR.ORDER))
THEN RAISE EXCEPTION 'Номер заказа не обнаружен %', id_ord
USING HINT = 'Проверьте номер заказа';
ELSE
RETURN (SELECT SUM(drink_price)
FROM bar.ORDER
INNER JOIN bar.DRINK_IN_ORDER ON bar.ORDER.ORDER_ID = bar.DRINK_IN_ORDER.ORDER_ID
INNER JOIN bar.DRINK on bar.DRINK_IN_ORDER.DRINK_ID = bar.DRINK.DRINK_ID
WHERE bar.Order.ORDER_ID = id_ord);
END IF;
END;
$BODY$
LANGUAGE plpgsql;
```

```
select Order_price(2);


CREATE OR REPLACE FUNCTION Most_popular_drinks (day date)
RETURNS varchar(20)
AS $BODY$
BEGIN
IF (day NOT IN (SELECT order_date FROM BAR.ORDER))
THEN RAISE EXCEPTION 'Некорректная дата %', day
USING HINT = 'Проверьте дату заказа';
ELSE
RETURN (SELECT drink_nm
FROM (
    SELECT drink_nm, count(bar.order.order_id) as drink_counter
      FROM bar.ORDER
            INNER JOIN bar.DRINK_IN_ORDER ON bar.ORDER.ORDER_ID =
bar.DRINK_IN_ORDER.ORDER_ID
            INNER JOIN bar.DRINK on bar.DRINK_IN_ORDER.DRINK_ID = bar.DRINK.DRINK_ID
      WHERE bar.order.order_date = day
      group by drink_nm
      ) as addition
where drink_counter in (select max(drink_counter) from (
    SELECT drink_nm, count(bar.order.order_id) as drink_counter
      FROM bar.ORDER
            INNER JOIN bar.DRINK_IN_ORDER ON bar.ORDER.ORDER_ID =
bar.DRINK_IN_ORDER.ORDER_ID
            INNER JOIN bar.DRINK on bar.DRINK_IN_ORDER.DRINK_ID = bar.DRINK.DRINK_ID
      WHERE bar.order.order_date = day
      group by drink_nm
      ) as addition));
END IF;
END;
$BODY$
```

```
LANGUAGE plpgsql;

select Most_popular_drinks('2019-04-20');



CREATE OR REPLACE FUNCTION employee_update ()
RETURNS TRIGGER
AS $BODY$
BEGIN
UPDATE BAR.EMPLOYEE
  SET EXPERIENCE = EXPERIENCE + 1;
RETURN NEW;
END;
$BODY$
LANGUAGE plpgsql;

CREATE TRIGGER Emp_update
AFTER INSERT ON BAR.PAYMENT
FOR EACH ROW
EXECUTE PROCEDURE employee_update();



CREATE ROLE Director;
CREATE ROLE Manager;
CREATE ROLE Barman;

GRANT SELECT
ON bar.OFFICE
TO Director;
GRANT SELECT, INSERT, UPDATE
ON bar.EMPLOYEE
TO Director;
```

```
GRANT SELECT
ON bar.employee
TO Manager;
GRANT SELECT, INSERT, UPDATE
ON bar.ORDER
TO Manager;
GRANT SELECT, INSERT, UPDATE
ON bar.DRINK_IN_ORDER
TO Manager;
GRANT SELECT, UPDATE, INSERT
ON bar.DRINK
TO Barman;

select *
 from information_schema.tables
where table_schema = 'bar'
;
```