

Итоговый проект. Отчет

База данных турагентства

Смешкова Екатерина, Пахолкова Мария

1) Описание базы данных:

База данных предназначена для управления деятельностью туристического агентства. Работа агентства устроена следующим образом:

У сотрудников (employee) агентства есть четкая иерархия внутри компании: среди них есть начальники и подчиненные, одни сотрудники руководят другими (leads).

Клиент (client) обращается к сотруднику (employee) и они совершают (make) бронирование (booking).

Бронирование происходит (book) из числа доступных туров (tours).

Если у клиента забронирован тур, то ему назначается (assigned) трансфер (Transfer) от аэропорта до отеля.

После посещения тура клиент может оставить (leave) один или несколько отзывов (review) о (about) своей поездке.

Более подробно про сущности:

В таблице Tour хранится информация о доступных турах: id тура (первичный ключ), место направления, длительность и цена тура.

В таблице Client хранится информация о клиентах турагентства: id клиента (первичный ключ), имя и почта.

В таблице Employee хранится информацию о сотрудниках (id, имя и почта), их должностях и иерархических связях.

В таблице Booking фиксируются детали бронирований, связывая клиентов (id клиента), туры (id тура) и сотрудников (id сотрудника), что позволяет отслеживать платежи (оплачен ли тур или нет) и даты начала туров. Нумерация броней в турагентстве включает в себя номер бронирования и дату начала тура, так как каждый день id обновляется и заново начинается с единицы.

В таблице Transfer хранится информация о машине, которая сопровождает клиентов на протяжении их поездки. При этой одна машина может быть прикреплена к нескольким турам, поэтому первичный ключ составной и включает в себя номер машины, дату начала тура и id бронирования. Transfer назначается только в том случае, если тур был оплачен.

В таблице Review хранятся отзывы клиентов о туре (оценка от 1 до 5), первичный ключ является составным и содержит информацию о id отзыва, id тура и id клиента. Это сделано для того, чтобы один и тот же клиент агентства мог оставлять несколько отзывов об одном или разных турах, которые он посетил.

2) ER-диаграмма:

Tour

{id_tour: int, destination: string, duration: int, price: decimal]}

Client

{id_client: int, name: string, email: string]}

Employee

{id_emp: int, name: string, email: string, post: string]}

Booking

{id_booking: int, start_date: date, **id_client**: int, **id_tour**: int, **id_emp**: int, payment: string]}

Review

{id_review: int, **id_tour**: int, **id_client**: int, grade: int]}

Transfer

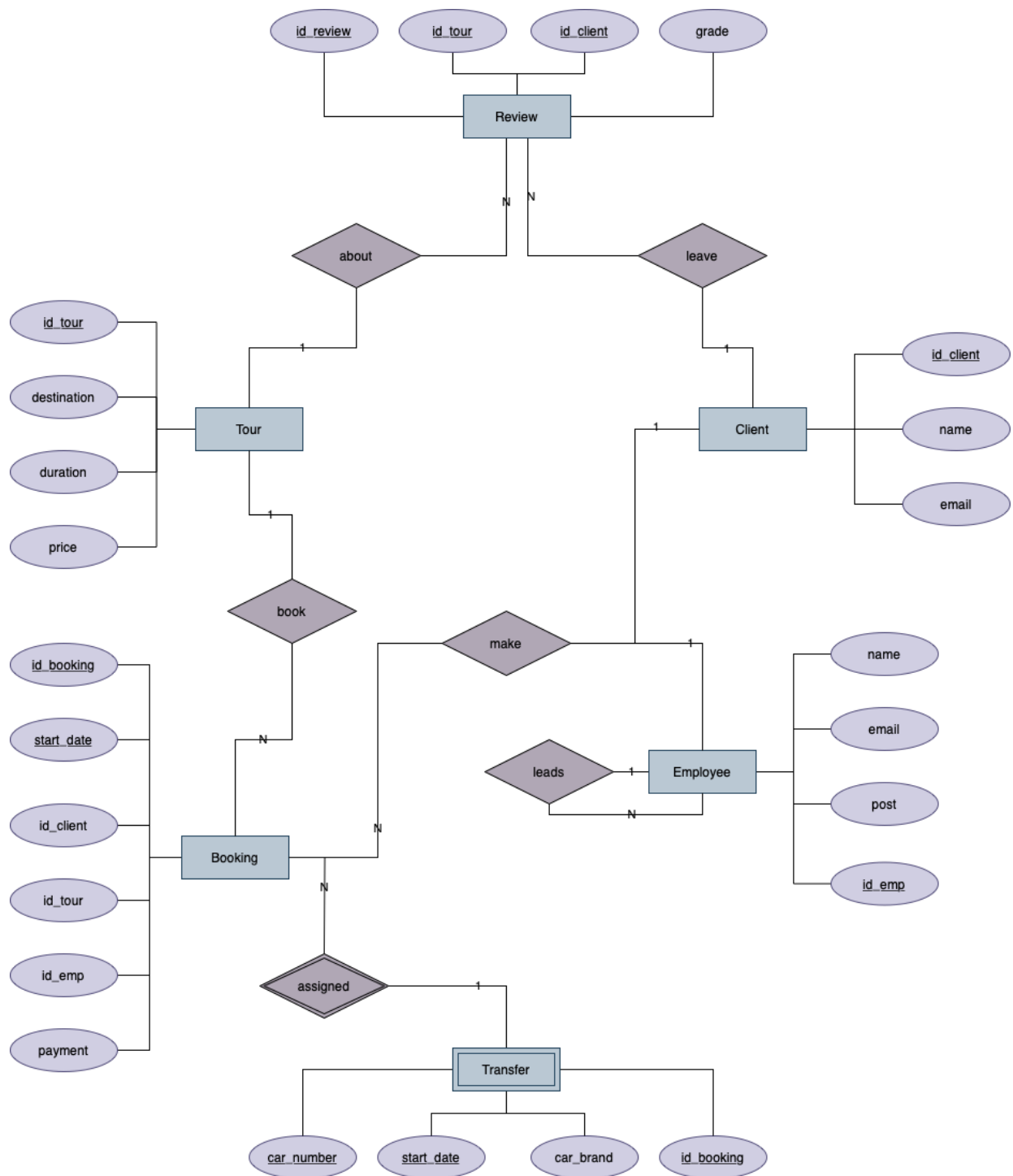
{car_number: string, car_brand: string, **id_booking**: int, **start_date**: date]}

Leads

{**id_boss**: int, **id_emp**: int]}

Make

{**id_client**: int, **id_emp**: int, **id_booking**: int, **start_date**: date]}



3) SQL код:

-- Создаем таблицы

```

CREATE TABLE Tour (
  id_tour INT PRIMARY KEY,
  destination VARCHAR(50) NOT NULL,

```

```
    duration INT NOT NULL CHECK (duration BETWEEN 1 AND 365),  
    price DECIMAL(10, 2) NOT NULL CHECK (price > 0)  
);
```

```
CREATE TABLE Client (  
    id_client INT PRIMARY KEY,  
    name VARCHAR(100) NOT NULL,  
    email VARCHAR(50) NOT NULL UNIQUE  
);
```

```
CREATE TABLE Employee (  
    id_emp INT NOT NULL PRIMARY KEY,  
    name VARCHAR(255) NOT NULL,  
    email VARCHAR(255) NOT NULL UNIQUE,  
    post VARCHAR(255) NOT NULL,  
    id_boss INT DEFAULT NULL REFERENCES Employee(id_emp)  
);
```

```
CREATE TABLE Booking (  
    id_booking INT NOT NULL,  
    start_date DATE NOT NULL,  
    id_client INT NOT NULL,  
    id_tour INT NOT NULL,  
    id_emp INT,  
    payment VARCHAR(20) NOT NULL CHECK (payment IN ('YES', 'NO')),  
    CONSTRAINT pk_booking PRIMARY KEY (id_booking, start_date),  
    FOREIGN KEY (id_client) REFERENCES Client(id_client),  
    FOREIGN KEY (id_tour) REFERENCES Tour(id_tour),  
    FOREIGN KEY (id_emp) REFERENCES Employee(id_emp)  
);
```

```
CREATE TABLE Review (  
    id_review INT NOT NULL,  
    id_tour INT NOT NULL,  
    id_client INT NOT NULL,  
    grade INT NOT NULL CHECK (grade BETWEEN 1 AND 5),  
    CONSTRAINT pk_review PRIMARY KEY (id_review, id_tour, id_client),  
    FOREIGN KEY (id_tour) REFERENCES Tour(id_tour),  
    FOREIGN KEY (id_client) REFERENCES Client(id_client)  
);
```

```
CREATE TABLE Transfer (  
    car_number VARCHAR(20) NOT NULL,
```

```
car_brand VARCHAR(255),
id_booking INT NOT NULL,
start_date DATE NOT NULL,
CONSTRAINT pk_transfer PRIMARY KEY (car_number, id_booking, start_date),
FOREIGN KEY (id_booking, start_date) REFERENCES Booking(id_booking,
start_date)
);
```

-- Заполняем таблицы данными

```
INSERT INTO Tour (id_tour, destination, duration, price) VALUES
(1, 'Париж', 7, 108000.00),
(2, 'Рим', 5, 90000.00),
(3, 'Лондон', 3, 720000.00),
(4, 'Нью-Йорк', 7, 1305000.00),
(5, 'Токио', 10, 180000.00),
(6, 'Сидней', 8, 162000.00),
(7, 'Барселона', 6, 317000.00),
(8, 'Берлин', 4, 521000.00),
(9, 'Дубай', 5, 99000.00),
(10, 'Москва', 7, 126000.00);
```

```
INSERT INTO Client (id_client, name, email) VALUES
(1, 'Иванова Дарья Анатольевна', 'dashh_iv@gmail.com'),
(2, 'Петров Борис Викторович', 'boris_tiger@gmail.com'),
(3, 'Сидоров Андрей Михайлович', 'andrey_sidorov@gmail.com'),
(4, 'Кузнецов Дмитрий Сергеевич', 'kuzznetsovv01@gmail.com'),
(5, 'Смирнова Елена Ивановна', 'lena_smir@gmail.com'),
(6, 'Попов Фёдор Александрович', 'fedor.popov@gmail.com'),
(7, 'Лебедева Галина Петровна', 'galina_lebedeva_28@gmail.com'),
(8, 'Новиков Никита Владимирович', 'smile.nikita@gmail.com'),
(9, 'Соколова Ирина Юрьевна', 'irshkaaaa@gmail.com'),
(10, 'Орлов Павел Павлович', 'orlov@gmail.com');
```

-- Сотрудники, не имеющие начальников

```
INSERT INTO Employee (id_emp, name, email, post, id_boss) VALUES
(1, 'Козлов Иван Петрович', 'ivan_boss@yandex.ru', 'Руководитель агентства',
NULL),
(2, 'Морозова Анна Ивановна', 'anna.morozova@gmail.com', 'Руководитель
отдела финансов и маркетинга', NULL);
```

-- Вставка сотрудников, подчиненных начальникам

```
INSERT INTO Employee (id_emp, name, email, post, id_boss) VALUES
```

(3, 'Воробьев Николай Сергеевич', 'vorobey@gmail.com', 'Младший
туроператор', 1), -- Воробьев подчиняется Козлову
(4, 'Фролова Ольга Дмитриевна', 'olechka_frl@gmail.com', 'Старший
туроператор', 1), -- Фролова подчиняется Козлову
(5, 'Зайцев Алексей Викторович', 'zaytsev@gmail.com', 'Экскурсовод', 1), --
Зайцев подчиняется Козлову
(6, 'Ковалёва Марина Сергеевна', 'marina_kov_380@gmail.com', 'Контент-
менеджер', 2), -- Ковалёва подчиняется Морозовой
(7, 'Миронов Андрей Павлович', 'andrey_pavlovich@yandex.ru', 'Менеджер по
продажам', 2), -- Миронов подчиняется Морозовой
(8, 'Романова Елена Александровна', 'elena_r@gmail.com', 'Стажер', 7), --
Романова подчиняется Миронову
(9, 'Мартынов Дмитрий Николаевич', 'martynofff@gmail.com', 'Специалист по
визовой поддержке', 1), -- Мартынов подчиняется Козлову
(10, 'Савельева Инна Петровна', 'savv.inna@mail.ru', 'HR-специалист', 1); --
Савельева подчиняется Козлову

INSERT INTO Booking (id_booking, start_date, id_client, id_tour, id_emp, payment)
VALUES

(1, '2024-03-01', 1, 5, 3, 'YES'),
(2, '2024-03-01', 5, 8, 3, 'NO'),
(1, '2024-04-15', 2, 7, 4, 'YES'),
(2, '2024-04-15', 4, 6, 4, 'YES'),
(3, '2024-04-15', 3, 1, 3, 'YES'),
(1, '2024-05-25', 9, 4, 8, 'NO'),
(2, '2024-05-25', 1, 5, 3, 'YES'),
(1, '2024-06-12', 5, 8, 8, 'YES'),
(1, '2024-06-20', 7, 9, 4, 'NO'),
(1, '2024-07-05', 1, 8, 8, 'YES');

INSERT INTO Review (id_review, id_tour, id_client, grade) VALUES

(1, 5, 1, 5),
(2, 7, 2, 4),
(3, 6, 4, 3),
(4, 1, 3, 5),
(5, 5, 1, 2),
(6, 8, 5, 3),
(7, 8, 1, 5);

INSERT INTO Transfer (car_number, car_brand, id_booking, start_date) VALUES

```
('Д123MC', 'Toyota', 1, '2024-03-01'),  
( 'M345PC', 'Audi', 1, '2024-04-15'),  
( 'E789HP', 'BMW', 2, '2024-04-15'),  
( 'K012OP', 'Mercedes', 3, '2024-04-15'),  
( 'M345PC', 'Audi', 2, '2024-05-25'),  
( 'H678CT', 'Volkswagen', 1, '2024-06-12'),  
( 'O901YX', 'Honda', 1, '2024-07-05');
```

-- Запросы

-- Вывести список клиентов и количество их бронирований, отсортировать по убыванию количества бронирований

```
SELECT c.id_client, c.name, COUNT(b.id_booking) AS booking_count  
FROM Client c  
LEFT JOIN Booking b ON c.id_client = b.id_client  
GROUP BY c.id_client, c.name  
ORDER BY booking_count DESC;
```

-- Туры с ценой выше среднего

```
SELECT *  
FROM Tour  
WHERE price > (SELECT AVG(price) FROM Tour);
```

-- Найти ТОП-3 самых востребованных направлений, имеющих наибольшее количество бронирований

```
SELECT t.destination, COUNT(b.id_booking) AS booking_count  
FROM Tour t  
JOIN Booking b ON t.id_tour = b.id_tour  
GROUP BY t.destination  
ORDER BY booking_count DESC  
LIMIT 3;
```

-- Топ-5 самых дорогих туров

```
SELECT *  
FROM Tour  
ORDER BY price DESC  
LIMIT 5;
```

-- Средние оценки туров

```
SELECT t.id_tour, t.destination, AVG(r.grade) AS average_grade  
FROM Tour t  
JOIN Review r ON t.id_tour = r.id_tour  
GROUP BY t.id_tour, t.destination;
```

-- Найти все туры, по которым не было ни одного бронирования

```
SELECT t.*  
FROM Tour t  
LEFT JOIN Booking b ON t.id_tour = b.id_tour  
WHERE b.id_booking IS NULL;
```

-- Информация о клиентах, забронированные туры у которых начинаются
менее чем через месяц

-- и до сих пор не оплачены, а также о сотруднике, курирующем это
бронирование

```
SELECT b.id_booking, b.start_date, c.name, c.email AS client_name, t.destination,  
e.name AS employee_name, b.payment  
FROM Booking b  
JOIN Client c ON b.id_client = c.id_client  
JOIN Tour t ON b.id_tour = t.id_tour  
LEFT JOIN Employee e ON b.id_emp = e.id_emp  
WHERE CURRENT_DATE <= b.start_date AND b.start_date <= DATE('now', '+1  
month')  
AND b.payment = 'NO';
```

-- Топ 3 клиента, посетивших больше всего туров

```
SELECT c.id_client, c.name, c.email, COUNT(b.id_booking) AS paid_tours_count  
FROM Client c  
JOIN Booking b ON c.id_client = b.id_client  
WHERE b.payment = 'YES' AND b.start_date < CURRENT_DATE  
GROUP BY c.id_client, c.name, c.email  
ORDER BY paid_tours_count DESC  
LIMIT 3;
```

-- Клиенты, у которых были неоплаченные туры (прошедшие)

```
SELECT c.id_client, c.name, c.email, COUNT(b.id_booking) AS unpaid_tours_count  
FROM Client c  
JOIN Booking b ON c.id_client = b.id_client  
WHERE b.payment = 'NO' AND b.start_date < CURRENT_DATE  
GROUP BY c.id_client, c.name, c.email  
HAVING COUNT(b.id_booking) > 0;
```

-- Машины, назначенные на предстоящие туры

```
SELECT t.car_number, t.car_brand, t.start_date  
FROM Transfer t  
WHERE DATE(t.start_date) > CURRENT_DATE;
```

-- Найти сотрудника, у которого наибольшая доля оплаченных броней от всего
их количества


```
SELECT id_emp, name
FROM Employee
WHERE id_emp IN (
    SELECT id_emp
    FROM Booking
    GROUP BY id_emp
    ORDER BY (CAST(SUM(CASE WHEN payment = 'YES' THEN 1 ELSE 0 END)
AS FLOAT) / COUNT(*)) * 100 ASC
    LIMIT 1
);
```

-- В связи с изменением цен на увеличить стоимость путевок в Париж и Лондон на 10%

```
UPDATE Tour
SET price = price*1.1
WHERE destination = 'Париж' AND destination = 'Рим';
```

-- Вывести марки всех автомобилей, у которых в названии есть буква 'a'

```
SELECT DISTINCT car_brand
FROM Transfer
WHERE car_brand LIKE '%a%'
ORDER BY car_brand ASC;
```

-- Вывести id топ-3 самых коротких туров

```
SELECT id_tour, MIN(duration) AS min_duration
FROM Tour
GROUP BY id_tour
ORDER BY min_duration
LIMIT 3;
```

-- Найти клиентов, которые один раз съездили на отдых более 3 месяцев назад, а после не делали повторных бронирований

```
SELECT id_client, name, email
FROM Client
WHERE id_client IN (
    SELECT id_client
    FROM Booking
    WHERE payment = 'YES' AND start_date <= DATE('now', '-3 month')
    GROUP BY id_client
    HAVING COUNT(id_booking) = 1
);
```