

**Московский авиационный институт
(национальный исследовательский университет)
Кафедра 806 «Вычислительная математика и
программирование»**

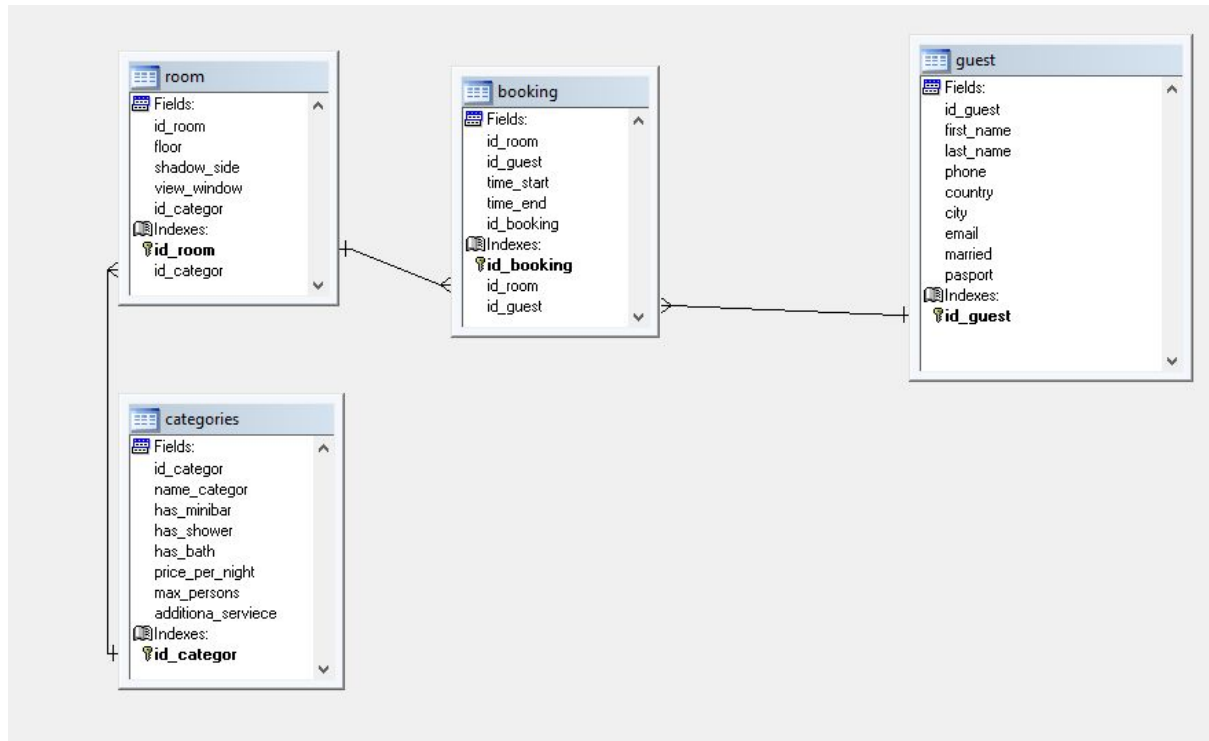
“База данных гостиницы”

**Работу выполнил: Сыроежкин К.Г
Группа: М8О-304Б-18
Телефон: +79997144729
email: kircatle@gmail.com**

Москва 2020 г

1. База данных в FoxPro

База данных гостиницы, в которой хранится информация о номерах, гостях проживающих в данном заведении, а также брони.



Содержание таблиц

Таблица 1. *rooms* (комнаты)

Id_room	Floor	Shadow_side	View_window	Id_categor
1	1	F	Good	2
2	1	F	Bad	1
3	2	T	Very good	3
4	2	T	Normal	2
5	3	F	Bad	4
6	3	T	Good	3
7	3	F	Good	3
8	4	T	Bad	1
9	4	T	Bad	1
10	4	T	Bad	4

В данной таблице содержится основная информация о комнатах гостиницы.

Поля таблицы:

- id_room - номер комнаты (тип - numeric);
- floor - этаж (тип - numeric);
- shadow_side - на темной ли стороне находится комната (тип - logical);

- view_window - вид из окна (тип - character);
- id_categor - номер категории (тип - numeric);

Таблица 2. categories (категории)

Id_categor	Name_categor	Has_minibar	Has_shower	Has_bath	Price_per_night	Max_persons	Additional_service
1	standart	F	F	F	30.000	1	F
2	standart+	F	F	F	40.000	2	T
3	Premium	F	T	F	50.000	2	T
4	Lux	F	T	T	75.000	3	T
5	Lux+	T	T	T	100.000	3	T

В данной таблице содержится информация о категориях комнаты (standart, lux и т.д.)

Поля таблицы:

- id_categor - номер категории (тип - numeric);
- Name_categor - название категории (тип - character);
- has_minibar - есть ли мини бар (тип - logical);
- has_shower - есть ли душ (тип - logical);
- has_bath - есть ли ванна (тип - logical);
- price_per_night - цена за ночь (тип - numeric);
- max_persons - максимальное количество человек (тип - numeric);
- additional_service - есть ли доп сервис (тип - logical);

Таблица 3. booking (бронь)

Id_room	Id_guest	Time_start	Time_end	Id_booking
5	1	02/22/20 01:11:11 AM	02/25/20 01:11:01 AM	1
5	5	02/22/20 01:11:11 AM	02/25/20 01:11:01 AM	2
5	6	02/22/20 01:11:11 AM	02/25/20 01:11:01 AM	3
4	3	04/11/20 07:00:00 AM	04/12/20 07:00:00 AM	4
4	4	04/11/20 07:00:00 AM	04/12/20 07:00:00 AM	5
1	2	05/12/20 02:20:00 AM	05/30/20 02:20:00 AM	6
10	4	06/12/20 02:20:00 AM	06/20/20 02:20:00 AM	7
2	3	06/12/20 02:20:00 AM	06/13/20 02:20:00 AM	8
3	3	06/12/21 02:20:00 AM	06/13/21 02:20:00 AM	9
3	7	11/12/21 12:00:00 AM	11/17/21 12:00:00 AM	10
6	8	11/12/21 12:00:00 AM	11/17/21 12:00:00 AM	11
7	8	12/12/21 12:00:00 AM	12/27/21 12:00:00 AM	12
9	3	01/12/22 12:00:00 AM	01/20/22 12:00:00 AM	13
9	3	01/22/22 12:00:00 AM	01/30/22 12:00:00 AM	14
9	3	03/22/22 12:00:00 AM	03/23/22 12:00:00 AM	15
8	7	04/22/22 12:00:00 AM	04/25/22 12:00:00 AM	16

Таблица содержит информацию о всех когда-либо снятых номерах

Поля таблицы:

- id_room - номер комнаты (тип - numeric);
- id_guest - идентификационный номер гостя (тип - numeric);
- time_start - начало съема (тип - datetime);
- time_end - конец съема (тип - datetime);
- id_booking - номер брони (тип - numeric);

Таблица 4. *guest* (гость)

Id_guest	First_name	Last_name	Phone	Country
1	Denis	Novikov	+79999999999	Russian
2	Oleg	Shewzhov	+38099993999	Ukrain
3	John	Vest	+13214151255	USA
4	Endi	Mitcher	+41241242151	British
5	Anton	Kozlov	+78980924907	Russian
6	Igor	Pivovarov	+78124789174	Russian
7	Efim	Chanken	+9012384091784	Lalalend
8	Dina	Laedf	+8321948910248	Halvsa

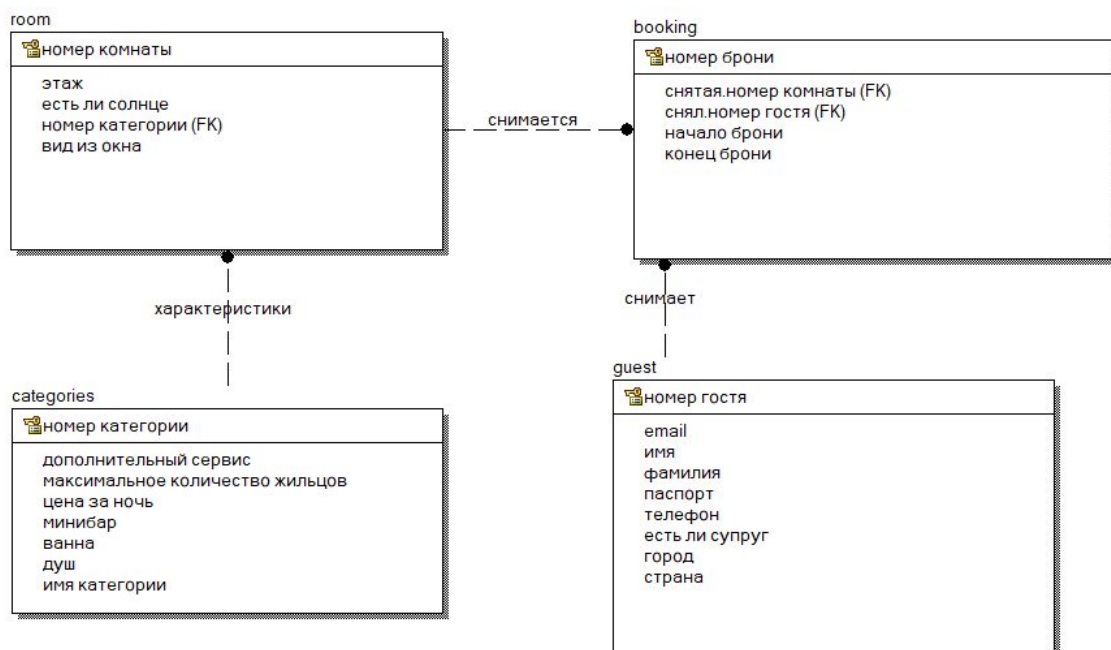
City	Email	Married	Pasport
Moscow	help_me@yandex.ru	T	451235235
Kiev	endi@yandex.ru	F	125124124
New-York	hagdsar@gmail.com	F	124141255
London	hella2@mail.com	T	125123445
Moscow	endar@mail.ru	T	215125254
Moscow	pivo_love@mail.ru	T	514124123
HEnki	i_love_you@icloud.com	F	412424124
Lagfssa	End_for_exile@mail.com	T	123124124

Таблица содержит всю информацию о госте, которая необходима гостинице для взаимодействия с ним

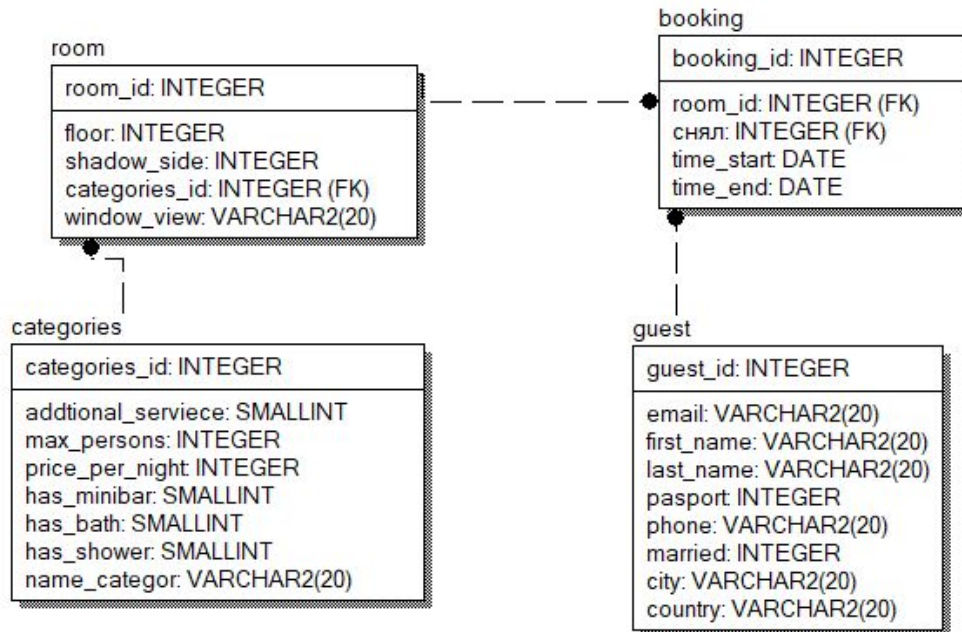
Поля таблицы:

- id_guest - номер гостя (тип - numeric);
- first_name - имя гостя (тип - character);
- last_name - фамилия гостя (тип - character);
- phone - мобильный телефон гостя (тип - character);
- country - страна проживания гостя (тип - character);
- city - город проживания гостя (тип - character);
- email - электронная почта (тип - character);
- married - замужем/женат (тип - logical);
- pasport - серия и номер паспорта (тип - numeric);

2. Логический уровень в ErWin



3. Физический уровень в ErWin



4. Сгенерированная БД

CREATE TABLE booking

```
(
    room_id          INTEGER NOT NULL ,
    booking_id       INTEGER NOT NULL ,
    снял             INTEGER NOT NULL ,
    time_start       DATE NULL ,
    time_end         DATE NULL
);
```

CREATE TABLE categories

```
(
    categories_id    INTEGER NOT NULL ,
    additional_service SMALLINT NULL ,
    max_persons      INTEGER NULL ,
    price_per_night  INTEGER NULL ,
    has_minibar      SMALLINT NULL ,
    has_bath         SMALLINT NULL ,
    has_shower       SMALLINT NULL ,
    name_categor     VARCHAR2(20) NULL
);
```

);

CREATE TABLE guest

```
(
    email          VARCHAR2(20) NULL ,
    guest_id       INTEGER NOT NULL ,
    first_name     VARCHAR2(20) NULL ,
    last_name      VARCHAR2(20) NULL ,
    passport       INTEGER NULL ,
    phone          VARCHAR2(20) NULL ,
    married        INTEGER NULL ,
    city           VARCHAR2(20) NULL ,
    country        VARCHAR2(20) NULL
);
```

CREATE TABLE room

```
(
    room_id        INTEGER NOT NULL ,
    floor          INTEGER NULL ,
    shadow_side     INTEGER NULL ,
    categories_id   INTEGER NOT NULL ,
    window_view     VARCHAR2(20) NULL
);
```

5. Сложные запросы

- Пары комнат заданной категории с одинаковым видом

```
SELECT r1.id_room as room1, r2.id_room as room2, r1.view_window as view,
r1.id_categor as category;
FROM room as r1, room as r2;
WHERE r1.view_window = r2.view_window AND r2.id_room > r1.id_room AND
r1.id_categor = r2.id_categor AND r1.id_categor = 1
```

Room1	Room2	View	Category
2	8	Bad	1
2	9	Bad	1
8	9	Bad	1

- Номера забронированные на срок выше среднего

```
SELECT SUM(booking.time_end-booking.time_start)/COUNT(booking.time_end) as
mean_time FROM booking INTO CURSOR tmp
SELECT booking.id_room as room, booking.time_start as start, booking.time_end as
end FROM booking;
WHERE (booking.time_end - booking.time_start) > tmp.mean_time
```

Room	Start	End
1	05/12/20 02:20:00 AM	05/30/20 02:20:00 AM
10	06/12/20 02:20:00 AM	06/20/20 02:20:00 AM
3	11/12/21 12:00:00 AM	11/17/21 12:00:00 AM
6	11/12/21 12:00:00 AM	11/17/21 12:00:00 AM
7	12/12/21 12:00:00 AM	12/27/21 12:00:00 AM
9	01/12/22 12:00:00 AM	01/20/22 12:00:00 AM
9	01/22/22 12:00:00 AM	01/30/22 12:00:00 AM

- Номера забронированные на максимальную сумму

```
SELECT b.id_room as room, c.price_per_night*ROUND((b.time_end -
b.time_start)/86400, 0) as price, ROUND((b.time_end - b.time_start)/86400, 0) as days,
r.id_categor as categor;
FROM booking as b, room as r, categories as c;
WHERE b.id_room = r.id_room AND c.id_categor = r.id_categor INTO cursor tmp
SELECT room, price, days FROM tmp;
WHERE price in (SELECT MAX(price) FROM tmp)
```

Room	Price	Days
7	750.000	15

- Определить постояльцев, которые никогда не бронировали номера данной категории

```
SELECT DISTINCT b.id_guest;
FROM booking as b;
WHERE b.id_guest NOT IN (SELECT b.id_guest FROM booking as b, room as r
WHERE b.id_room = r.id_room AND r.id_categor = 1)
```

Id_guest
1
2
4
5
6
8

- Определить наиболее востребованный тип номеров (бронируют выше среднего)

```

SELECT r.id_categor as categor, COUNT(r.id_categor) as count_categor FROM room
as r, booking as b;
WHERE b.id_room = r.id_room;
GROUP BY r.id_categor INTO CURSOR tmp1
SELECT SUM(tmp1.count_categor)/COUNT(tmp1.count_categor) as averegeRent
FROM tmp1 INTO CURSOR tmp2
SELECT tmp1.categor as id_categor, tmp1.count_categor as count_rent FROM tmp1,
tmp2;
WHERE tmp1.count_categor > tmp2.averegeRent

```

Id_categor	Count_rent
1	5

- Определить номер, который на заданном интервале дат бронировало наибольшее количество людей

```

SELECT b.id_room as room, COUNT(b.id_room) as count_rent FROM booking as b;
WHERE b.time_start between {^2020/04/10} AND {^2020/05/31} AND b.time_end
between {^2020/04/10} AND {^2020/05/31};
GROUP BY b.id_room INTO CURSOR tmp
SELECT tmp.room, tmp.count_rent FROM tmp;
WHERE tmp.count_rent in (SELECT MAX(tmp.count_rent) FROM tmp)

```

Room	Count_rent
4	2