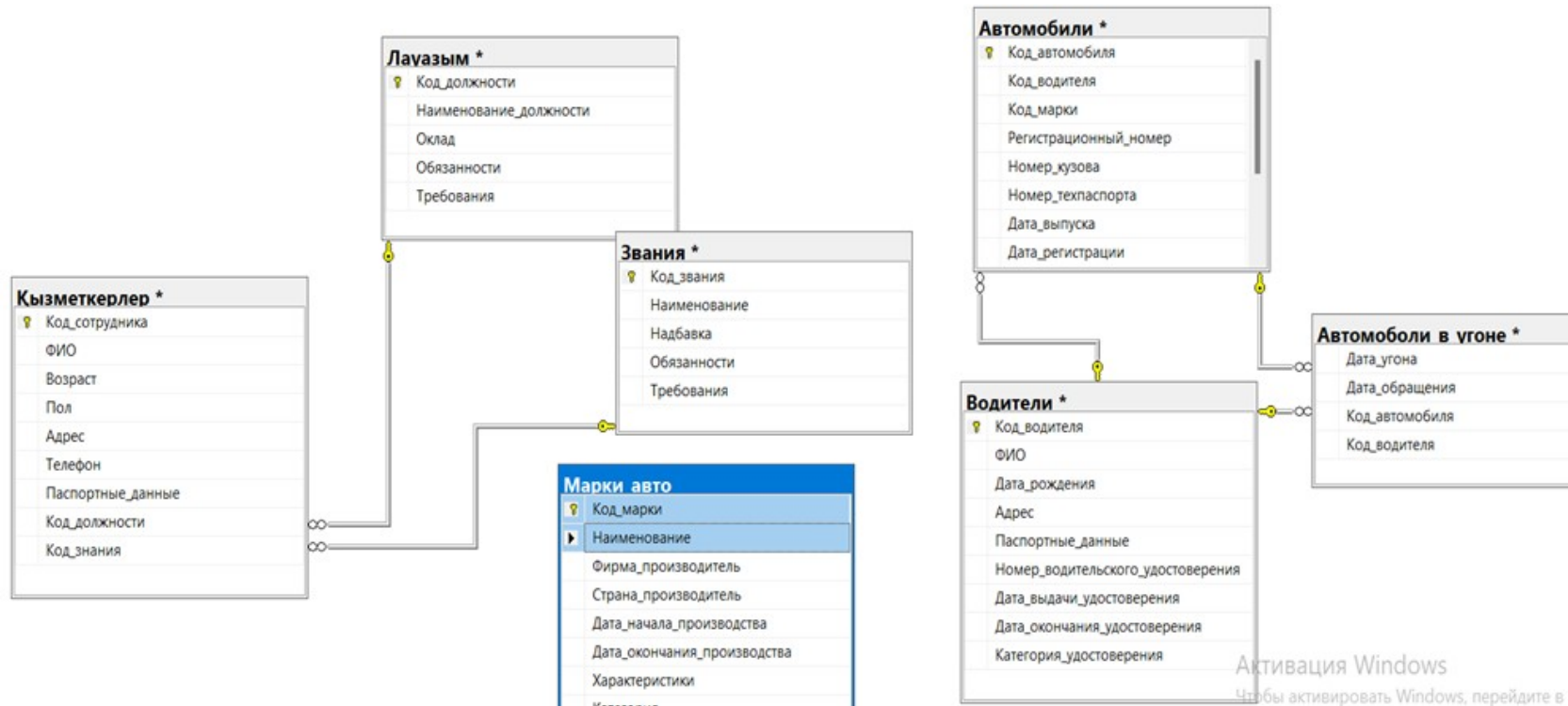


Задача состоит в создании базы данных «ГИБДД» для информационно-справочной системы которая будет обеспечивать хранение данных о расписании приема врачей и записи пациентов на прием. База данных должна учитывать особенности работы поликлиники, такие как:

врачи специалисты и участковые врачи, расписания врачей, которые не меняются на протяжении недели, работа врачей в две смены, разбиение приема на интервалы, кабинеты, в которых работают врачи в разное время, учет предоставленных услуг, медицинская страховка пациентов.

Информационная система должна предоставлять возможность пациентам записаться на прием к врачу, а также врачам просмотреть очередность записанных пациентов. Администрация должна иметь доступ к данным о количестве пациентов, записавшихся и пришедших на прием, а также к сведениям о загрузке кабинетов и врачей.

# ER diagram



Активация Windows  
Чтобы активировать Windows, перейдите в ра

# Таблицы

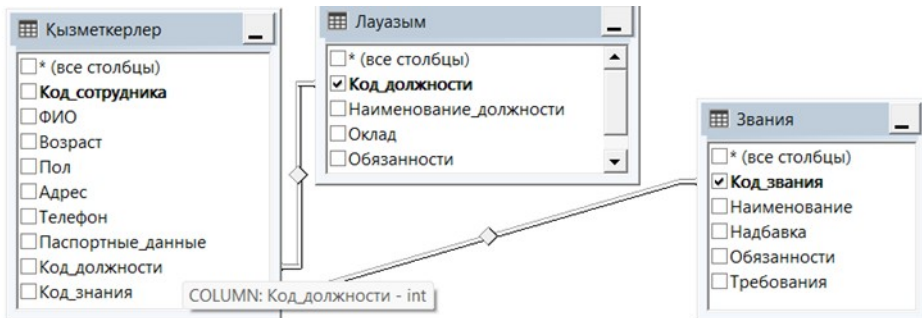
XE5088.ГИБДД - d...томобили\_в\_угоне

Дата_угона	Дата_обра...	Код_автом...	Код_водит...
2012-09-09	2012-09-09	112333	561316
2012-10-10	2012-10-10	313213	231644
2012-09-20	2012-09-20	646466	494562
2012-11-04	2012-11-04	649847	123131
2012-11-11	2012-11-11	897812	554645

XE5088.ГИБДД - dbo.Автомобили

Код_автом...	Код_водит...	Код_марки	Регистрац...	Номер_куз...	Номер_тех...	Дата_выпу...	Дата_реги...	Цвет	Техническ...	Дата_техн...	Описание	Код_сотру...
112333	489455	111111	8977	689	98888	2012-01-26	2012-02-26	Синий	R978XO116...	2012-03-26	Универсал	234
156464	314646	123456	5788	989	74511	2010-11-12	2010-12-12	Красный	R223XO545...	2011-01-12	Грузовик	645
231321	898965	215654	8788	549	56494	1996-02-06	1996-02-09	Серый	R223XO886...	2011-05-12	Универсал	234
313213	494562	215654	9788	963	89444	1995-01-19	1995-01-22	Синий	R511XO116...	2010-01-18	Вседорожн...	645
316416	994561	345548	8797	456	48123	2002-12-13	2002-12-18	Белый	R253XO116...	2009-09-12	Вседорожн...	894
465123	561316	457891	4544	123	99844	1990-08-19	1990-08-26	Черный	R223XO964...	2011-01-12	Универсал	987
498898	554645	546412	8974	789	99845	1962-10-06	1962-10-19	Желтый	R223XO164...	2011-01-12	Универсал	234
564196	649987	645511	4565	951	21611	1993-06-12	1993-06-25	Пурпурный	R223XO548...	2011-01-12	Универсал	897
646466	645645	798451	5649	753	23213	1991-02-02	1991-02-27	Черный	R223XO632...	2011-01-12	Грузовик	897
649847	996345	846551	4445	103	16555	1996-11-13	1996-11-16	Белый	R223XO999...	2011-01-12	Вседорожн...	897
879421	123131	945612	1232	232	45577	2001-06-11	1996-06-12	Черный	K223XO116...	2011-01-12	Универсал	232
897789	489898	945612	5499	302	89849	2000-07-11	2000-07-22	Красный	R999XO116...	2011-01-12	Универсал	645
897812	162312	945612	9978	309	44994	2002-08-11	2002-08-26	Зеленый	R223XO116...	2011-01-12	Вседорожн...	894
964132	564649	945612	4974	507	16544	2003-12-11	2003-12-27	Синий	R223XO956...	2011-01-12	Вседорожн...	894
987852	231644	945612	1133	907	84941	2011-01-11	2011-01-28	Черный	R554XO116...	2011-01-12	Вседорожн...	988

Код_сотру...	Код_марки	Код_водит...	ФИО	Фирма_пр...
234	111111	489455	Музаффар...	Бугатти
645	123456	314646	Нурланова ...	УАЗ
234	215654	898965	Музаффар...	ЛАДА
645	215654	494562	Нурланова ...	ЛАДА
894	345548	994561	Музаффар ...	УАЗ
987	457891	561316	Ильин Вла...	ТагАЗ
234	546412	554645	Музаффар...	ЗИЛ
897	645511	649987	Сайлаубае...	УАЗ
897	846551	996345	Сайлаубае...	Тойота
232	945612	123131	Калдыбаев ...	Тойота
645	945612	489898	Нурланова ...	Тойота
894	945612	162312	Музаффар ...	Тойота
894	945612	564649	Музаффар ...	Тойота
988	945612	231644	Сыздыкова...	Тойота



```

SELECT dbo.Лауазым.Код_должности, dbo.Звания.Код_звания
FROM   dbo.Кызметкерлер INNER JOIN
       dbo.Лауазым ON dbo.Кызметкерлер.Код_должности = dbo.Лауазым.Код_должности INNER JOIN
       dbo.Звания ON dbo.Кызметкерлер.Код_звания = dbo.Звания.Код_звания

```

Код_долж...	Код_звания
3333	555
1111	444
3333	333
4444	444
5555	555
5555	555
2222	111
2222	222
3333	222
4444	333

Analysis Services Scripts1 - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Execute

Object Explorer

Connect

MDXQuery1.mdx - not connected\* DMXQuery1.dmx - not connected

Cube: Not connected.

Metadata Functions

Search Model

Measure Group:

Error loading metadata: Not connected.

```

USE policlinika;

CREATE TABLE if not exists Doctors (
    Doctor_ID INT NOT NULL AUTO_INCREMENT,
    Last_Name VARCHAR(50) NOT NULL,
    First_Name VARCHAR(50) NOT NULL,
    Middle_Name VARCHAR(50),
    Specialization VARCHAR(100) NOT NULL,
    Is_Local_Doctor BOOLEAN NOT NULL,
    Local_Doctor_Area_Number INT,
    Week_Schedule_ID INT NOT NULL,
    PRIMARY KEY (Doctor_ID)
);

CREATE TABLE Local_Doctor_Area (
    Area_Number INT NOT NULL AUTO_INCREMENT,
    Streets VARCHAR(500) NOT NULL,
    PRIMARY KEY (Area_Number)
);
select * from Local_Doctor_Area;

CREATE TABLE Week_Schedule (
    Schedule_ID INT NOT NULL AUTO_INCREMENT,
    Doctor_ID INT NOT NULL,
    Weekday ENUM('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday') NOT NULL,
    Morning_Start_Time TIME NOT NULL,
    Morning_End_Time TIME NOT NULL,
    Evening_Start_Time TIME NOT NULL,
    Evening_End_Time TIME NOT NULL,
    PRIMARY KEY (Schedule_ID),
    FOREIGN KEY (Doctor_ID) REFERENCES Doctors(Doctor_ID)
);
select * from Week_Schedule;

```

Solution Explorer

Search Solution Explorer (Ctrl+X)

Solution 'Analysis Services Scripts1' (1 pr...

- Analysis Services Scripts1
  - Connections
  - Queries
    - DMXQuery1.dmx
    - MDXQuery1.mdx
    - Miscellaneous

Result Grid

Filter Rows:

Edit: Export/Import: Wrap Cell Content:

Schedule_ID	Doctor_ID	Weekday	Morning_Start_Time	Morning_End_Time	Evening_Start_Time	Evening_End_Time
1	1	Monday	08:00:00	12:00:00	14:00:00	18:00:00
2	1	Tuesday	08:00:00	12:00:00	14:00:00	18:00:00
3	1	Wednesday	08:00:00	12:00:00	14:00:00	18:00:00
4	1	Thursday	08:00:00	12:00:00	14:00:00	18:00:00
5	1	Friday	08:00:00	12:00:00	14:00:00	18:00:00
6	1	Saturday	09:00:00	12:00:00	13:00:00	16:00:00
7	2	Monday	10:00:00	13:00:00	15:00:00	19:00:00
8	2	Tuesday	10:00:00	13:00:00	15:00:00	19:00:00
9	2	Wednesday	10:00:00	13:00:00	15:00:00	19:00:00
10	2	Thursday	10:00:00	13:00:00	15:00:00	19:00:00
11	2	Friday	10:00:00	13:00:00	15:00:00	19:00:00
*	NULL	NULL	NULL	NULL	NULL	NULL

Administration Schemas

Information

Schema: policlinika

Week\_Schedule 9 x

Apply Revert Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
36	00:00:29	select * from Local_Doctor_Area LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

### ***Ограничения целостности данных***

1 Ограничение NOT NULL: Это ограничение гарантирует, что в столбцах, помеченных как NOT NULL, не будет храниться NULL-значение.

2 Ограничение UNIQUE: Это ограничение гарантирует, что значения в указанных столбцах уникальны.

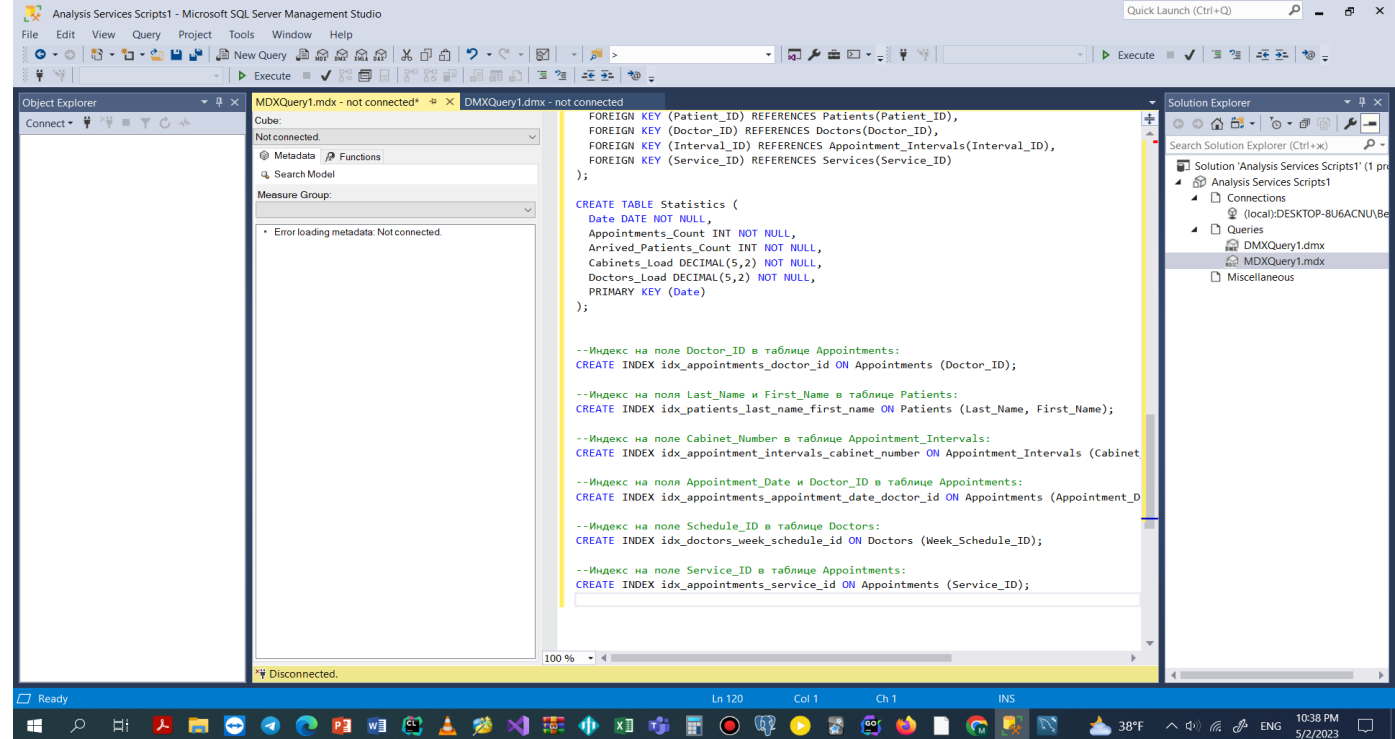
3 Ограничение PRIMARY KEY: Это ограничение гарантирует, что значения в указанном столбце уникальны и не могут быть NULL.

4 Ограничение FOREIGN KEY: Это ограничение гарантирует, что значения в указанном столбце ссылаются на значения в столбце первичного ключа другой таблицы.

5 Ограничение CHECK: Это ограничение гарантирует, что значения в указанном столбце соответствуют определенным условиям

# Индексы

- Индекс на поле Doctor\_ID в таблице Appointments:
- CREATE INDEX idx\_appointments\_doctor\_id ON Appointments (Doctor\_ID);
- Индекс на поля Last\_Name и First\_Name в таблице Patients:
- CREATE INDEX idx\_patients\_last\_name\_first\_name ON Patients (Last\_Name, First\_Name);
- Индекс на поле Cabinet\_Number в таблице Appointment\_Intervals:
- CREATE INDEX idx\_appointment\_intervals\_cabinet\_number ON Appointment\_Intervals (Cabinet\_Number);
- Индекс на поля Appointment\_Date и Doctor\_ID в таблице Appointments:
- CREATE INDEX idx\_appointments\_appointment\_date\_doctor\_id ON Appointments (Appointment\_Date, Doctor\_ID);
- Индекс на поле Schedule\_ID в таблице Doctors:
- CREATE INDEX idx\_doctors\_week\_schedule\_id ON Doctors (Week\_Schedule\_ID);
- Индекс на поле Service\_ID в таблице Appointments:
- CREATE INDEX idx\_appointments\_service\_id ON Appointments (Service\_ID);



```
FOREIGN KEY (Patient_ID) REFERENCES Patients(Patient_ID),
FOREIGN KEY (Doctor_ID) REFERENCES Doctors(Doctor_ID),
FOREIGN KEY (Interval_ID) REFERENCES Appointment_Intervals(Interval_ID),
FOREIGN KEY (Service_ID) REFERENCES Services(Service_ID)
);

CREATE TABLE Statistics (
    Date DATE NOT NULL,
    Appointments_Count INT NOT NULL,
    Arrived_Patients_Count INT NOT NULL,
    Cabinets_Load DECIMAL(5,2) NOT NULL,
    Doctors_Load DECIMAL(5,2) NOT NULL,
    PRIMARY KEY (Date)
);

--Индекс на поле Doctor_ID в таблице Appointments:
CREATE INDEX idx_appointments_doctor_id ON Appointments (Doctor_ID);

--Индекс на поля Last_Name и First_Name в таблице Patients:
CREATE INDEX idx_patients_last_name_first_name ON Patients (Last_Name, First_Name);

--Индекс на поле Cabinet_Number в таблице Appointment_Intervals:
CREATE INDEX idx_appointment_intervals_cabinet_number ON Appointment_Intervals (Cabinet_Number);

--Индекс на поля Appointment_Date и Doctor_ID в таблице Appointments:
CREATE INDEX idx_appointments_appointment_date_doctor_id ON Appointments (Appointment_Date, Doctor_ID);

--Индекс на поле Schedule_ID в таблице Doctors:
CREATE INDEX idx_doctors_week_schedule_id ON Doctors (Week_Schedule_ID);

--Индекс на поле Service_ID в таблице Appointments:
CREATE INDEX idx_appointments_service_id ON Appointments (Service_ID);
```

- Индекс на поле Area\_Number в таблице Local\_Doctor\_Area:
- CREATE INDEX idx\_local\_doctor\_area\_area\_number ON Local\_Doctor\_Area (Area\_Number);
- 
- Индекс на поле Doctor\_ID в таблице Week\_Schedule:
- CREATE INDEX idx\_week\_schedule\_doctor\_id ON Week\_Schedule (Doctor\_ID);
- 
- Индекс на поле Cabinet\_Name в таблице Cabinets:
- CREATE INDEX idx\_cabinets\_cabinet\_name ON Cabinets (Cabinet\_Name);
- 
- Индекс на поля Appointment\_Date и Is\_Free в таблице Appointments:
- CREATE INDEX idx\_appointments\_appointment\_date\_is\_free ON Appointments (Appointment\_Date, Is\_Free);



# триггеры для базы данных «ГИБДД»:

1 Триггер для автоматической генерации ID записей приема:

- CREATE TRIGGER trg\_generate\_appointment\_id
- BEFORE INSERT ON Appointments
- FOR EACH ROW
- BEGIN
- SET NEW.Appointment\_ID = CONCAT('AP', LPAD((SELECT COUNT(\*) + 1 FROM Appointments), 5, '0'));
- END

2 Триггер для удаления связанных записей при удалении врача:

- CREATE TRIGGER trg\_delete\_related\_data\_on\_doctor\_delete
- BEFORE DELETE ON Doctors
- FOR EACH ROW
- BEGIN
- DELETE FROM Week\_Schedule WHERE Doctor\_ID = OLD.Doctor\_ID;
- DELETE FROM Appointment\_Intervals WHERE Doctor\_ID = OLD.Doctor\_ID;
- DELETE FROM Appointments WHERE Doctor\_ID = OLD.Doctor\_ID;
- END
-

- 3 Триггер для обновления статистики приема пациентов при добавлении новой записи на прием:

- CREATE TRIGGER trg\_update\_statistics\_on\_appointment\_insert

- AFTER INSERT ON Appointments

- FOR EACH ROW

- BEGIN

- UPDATE Statistics SET

- Appointments\_Count = Appointments\_Count + 1,

- Cabinets\_Load = (

- SELECT

- COUNT(\*)

- FROM

- Appointments

- WHERE

- Appointment\_Date = NEW.Appointment\_Date

- AND Cabinet\_Number = NEW.Cabinet\_Number

- ),

- Doctors\_Load = (

- SELECT

- COUNT(\*)

- FROM

- Appointments

- WHERE

- Appointment\_Date = NEW.Appointment\_Date

- AND Doctor\_ID = NEW.Doctor\_ID

- )

- WHERE

- Date = NEW.Appointment\_Date;

- END