



**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ЯДЕРНЫЙ УНИВЕРСИТЕТ  
«МИФИ»**

---

**Институт интеллектуальных кибернетических систем**

**Кафедра № 22**

**Курсовая работа по моделированию предметных сред  
и проектированию баз данных**

**Вариант № 4 «База данных эмоций»**

Выполнил	Борисова А.А.
Группа	Б18-504
Вариант	4
Преподаватель	Климов В.В.
Проверяющий	
Оценка	

**Москва 2019**

# Содержание

1.	ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ .....	3
1.1.	Формулировка задания .....	3
1.2.	Конкретизация предметной области.....	3
1.3.	Пользователи системы.....	3
1.4.	Сроки хранения информации .....	4
1.5.	События, изменяющие состояние Базы Данных.....	4
1.6.	Основные запросы к Базе Данных (на естественном языке) .....	4
2.	КОНЦЕПТУАЛЬНО-ИНФОРМАЦИОННАЯ МОДЕЛЬ ПРЕДМЕТНОЙ ОБЛАСТИ .....	4
2.1.	ER-диаграмма модели (SilverRun) .....	4
2.2.	Оценка мощностных характеристик сущностей и связей.....	5
3.	КОНЦЕПТУАЛЬНОЕ ПРОЕКТИРОВАНИЕ .....	6
3.1.	Принятые проектные соглашения.....	6
3.2.	Используемые в системе кодификаторы .....	6
3.3.	Концептуальная модель Базы Данных (SilverRun) .....	7
4.	ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ .....	8
4.1.	ER-диаграмма Базы Данных (ERWin Logical).....	8
4.2.	Схемы отношений Базы Данных (ERWin Phisycal).....	8
4.3.	Схема реляционной Базы Данных .....	8
4.4.	Схемы основных запросов на реляционной алгебре .....	8
5.	ФИЗИЧЕСКОЕ ПРОЕКТИРОВАНИЕ .....	12
5.1.	Создание Базы Данных.....	12
5.2.	Создание Таблиц.....	14
5.3.	Заполнение таблиц.....	17
5.4.	Запросы в терминах SQL.....	17
5.5.	Оценка размеров Базы Данных и каждого из файлов .....	17
5.6.	Заполние базы данных.....	18

## **1. Описание предметной области**

### **1.1. Формулировка задания**

Спроектировать аннотированную реляционную базу данных эмоций на основе мимики. База данных должна позволять хранить набор изображений в диапазоне эмоций. Аннотированный медиа-контент о поведении человека может быть использован для обучения, тестирования и проверки алгоритмов разработки систем распознавания эмоций.

### **1.2. Конкретизация предметной области**

В предметную область входят графические представления эмоций.

Цель создания базы данных: хранение ссылок и информации на отсортированные данные для нейронной сети, обученной распознавать эмоциональный окрас данных.

Моделируемая ситуация: проводится исследование реакции людей на некоторый продукт, получается массив данных. Нейронная сеть сортирует массив фотографий исследования, результаты переносятся в базу данных, в которой можно вывести статистику и провести анализ. Это позволит собрать быстрый отклик на услугу/продукт и улучшить качества продукта.

Сущности:

1. Данные о людях, принявшие участие в исследовании. Только информация, нужная для анализа: возраст и пол.
2. Информация о конкретной фотографии: ссылка на эмоцию, представленную на фото; ссылка на файл (сами фотографии в БД хранить слишком ресурсозатратно); ссылка на человека (если собирались более персонализированные данные); ссылка на эксперимент, где задействуется фото.
3. Эксперимент. Дата проведения, ссылка на продукт. Можно провести несколько экспериментов с одним и тем же продуктом, не смешивая данные разных периодов.
4. Продукт: ID и имя. Таблица-кодификатор.
5. Список определённых эмоций. От более конкретных (счастье, страх и т.д.) до более общих (позитивные, негативные и т.д.). Также скорее является кодификатором, прикрепляющим к ID имя и ссылку к обобщающей категории, если такая есть.
- 6.

### **1.3. Пользователи системы**

- Куратор исследования может вносить новые данные: данные по эксперименту, данные участников, фото и т.п.
- Администратор может вносить новые классы эмоций, характеристики тестируемых и т.п.
- Статист может выполнять запросы к БД, проводить некоторую статистику.

#### 1.4. Сроки хранения информации

Основной костяк распределения эмоциональных реакций по типам сохраняется всегда. Неактуальные исследования могут удаляться или переноситься в удалённый архив на случай надобности в обучении нейросети. Данные для тренировки могут храниться в копии БД.

#### 1.5. События, изменяющие состояние Базы Данных

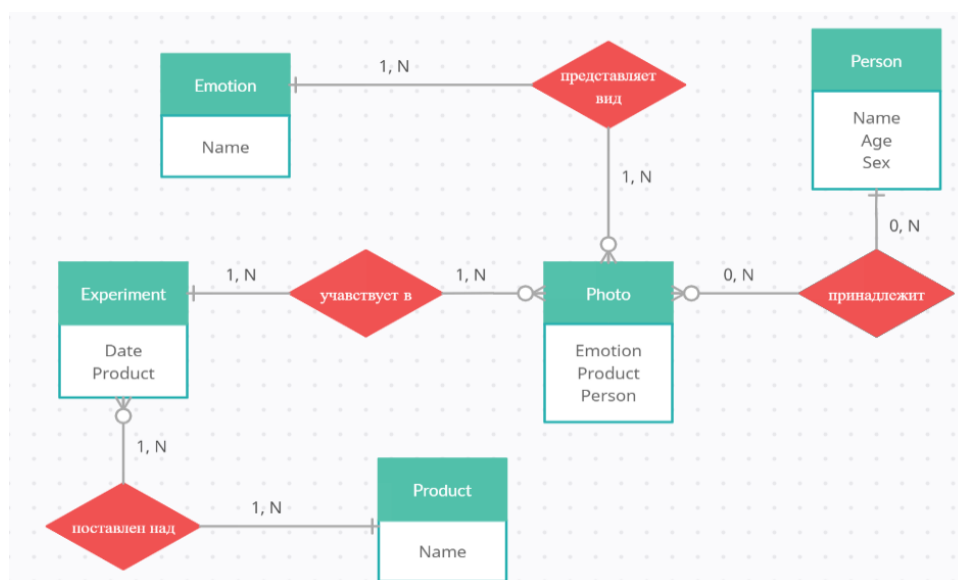
- Проведение нового эксперимента;
- Пополнение данных новыми участниками, требующими конкретной информации по ним;
- Поступление новых данных;
- Появление новых атрибутов сущностей;
- Новое обучение нейросети на дополнительных данных.

#### 1.6. Основные запросы к Базе Данных (на естественном языке)

1. Показать отсортированные данные последнего исследования.
2. реакции топ10 самых возрастных участников.
3. Показать реакции на продукты и процент позитивных от их числа.
4. Гистограмма по возрастным категориям.
5. Количество реакций по половому признаку на определённый продукт.

### 2. Концептуально-информационная модель предметной области

#### 2.1. ER-диаграмма модели



## 2.2. Оценка мощностных характеристик сущностей и связей

Приводится таблица следующего вида.

Сущность/связь	Мощность		
	минимальная	средняя	максимальная
Человек	100	500	10 000
Фотография	1000	10 000	50 000
Эмоция	5	10	20
Эксперимент	3	5	50
Продукт	3	5	20
	1	5	10
Человеку принадлежит фотография	0	3	20
На фотографии отображена эмоция	1	1	1
Эксперимент над данным продуктом	1	2	5
Фотография относится к эксперименту	1000	5000	30 000
Эмоция является дочерней для другой	1	1	1

Фотография может не иметь владельца, если для эксперимента важна только статистика реакций. Но может быть несколько фотографий одного человека как в разных, так и в одном эксперименте. На одной фотографии может быть только одна эмоция. Эмоция может быть прямо дочерней только для одной родительской.

Мощность сущности “эксперимент” зависит от количества продуктов. Мощности связи принадлежности фотографии эксперименту зависит от количества фотографий.

Для тренировки нейронной сети нужно несколько тысяч фотографий, как и для теста. Минимальная выборка людей для эксперимента будет 100, максимальное число зависит от масштаба эксперимента, вплоть до нескольких тысяч.

### 3. Концептуальное проектирование

#### 3.1. Принятые проектные соглашения

При переходе от КМПО к КМБД добавлены идентификаторы к каждой таблице. Делается новая связь в таблице “Emotion”: parent\_id ссылается на обобщающую эмоцию для некоторых пунктов. Решение выделить продукты как таблицу кодификаторов окончательно.

#### 3.2. Используемые в системе кодификаторы

Приводится описание использующихся кодификаторов – коротких таблиц типа «ключ-значение». Для каждого кодификатора необходимо указать место его хранения – отдельной сущностью в БД или в прикладном ПО.

Примеры:

Пол		ПО
Код	Значение	
0	Неизвестно	
1	Женский	
2	Мужской	

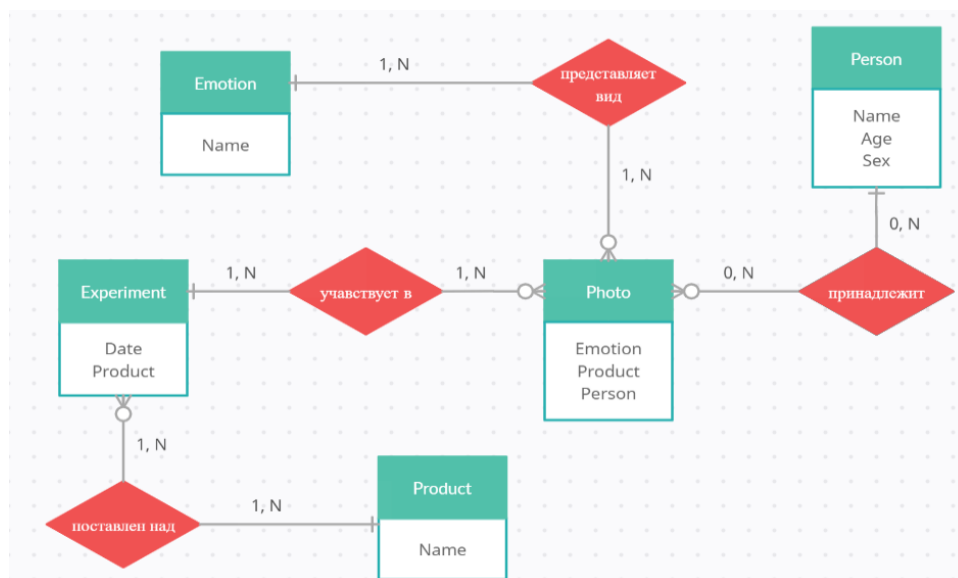
Храним в ПО, т.к. не может быть изменений в составе данных кодификатора.

Эмоции	БД
Код	Значение
1	Angry
2	Disgust
3	Fear
4	Happy
5	Neutral
6	Sad
7	Surprise
8	Weak
9	Strong
10	Positive
11	Negative

Продукт	БД
Код	Значение
1	Фильм
2	Напиток
3	Сырок
4	Сыр
5	Молоко

Храним в БД, т.к. могут быть изменения в составе данных.

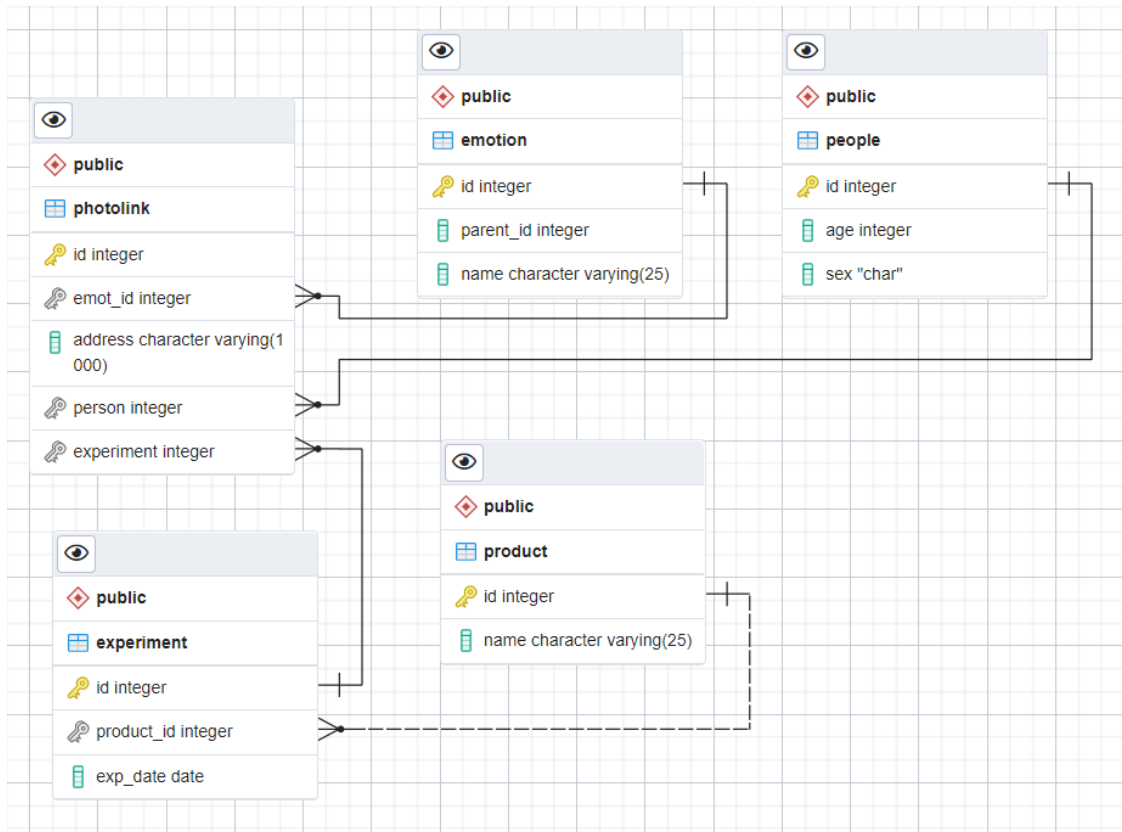
### 3.3. Концептуальная модель Базы Данных



## 4. Логическое проектирование

### 4.1. ER-диаграмма Базы Данных (ERWin Logical)

### 4.2. Схемы отношений Базы Данных (ERWin Phisycal)



### 4.3. Схема реляционной Базы Данных

Схема отношений:

R1	Photolink(#id, emot_id, address, person, experiment)
R2	Emotion(#id, parent_id, name)
R3	People(#id, age, sex)
R4	Experiment(#id, product_id, exp_date)
R5	Product(#id, name)

### 4.4. Схемы основных запросов на реляционной алгебре

1) Данные по последнему эксперименту

```
select ex.id as experiment_id, p.name AS product_name, ex.exp_date,  
ph.person, em.emot_name FROM experiment ex
```



```

join product p on (ex.product_id=p.id)
join photolink ph on (ex.id=ph.experiment)
join emotion em on (em.id=ph.emot_id)
WHERE ex.exp_date in (select max(ex.exp_date) from experiment ex)
order by person;

```

	experiment_id integer	product_name character varying (25)	exp_date date	person integer	emot_name character varying (25)
1	1	film	2021-06-18	1	surprise
2	3	drink	2021-06-18	1	surprise
3	3	drink	2021-06-18	1	angry
4	3	drink	2021-06-18	1	happy
5	3	drink	2021-06-18	1	angry
6	3	drink	2021-06-18	1	fear
7	3	drink	2021-06-18	1	sad
8	3	drink	2021-06-18	1	happy

2) Реакции топ10 самых возрастных участников

```

SELECT elders.id, age, sex, pr.name, ph.address
FROM product pr
join experiment ex on (ex.product_id=pr.id)
join photolink ph on (ph.experiment=ex.id)
join (select * from people
order by age desc
limit 10) as elders on (elders.id=ph.person)
order by pr.id desc;

```

	id integer	age integer	sex integer	name character varying (25)	address character varying (1000)
1	58	76	2	milk	D:/Media/Pics/emot/train\surp...
2	18	78	2	milk	D:/Media/Pics/emot/train\fear...
3	55	75	0	milk	D:/Media/Pics/emot/train\neut...
4	55	75	0	milk	D:/Media/Pics/emot/train\sad\...
5	21	77	2	milk	D:/Media/Pics/emot/train\neut...
6	29	79	1	milk	D:/Media/Pics/emot/train\surp...
7	49	76	2	milk	D:/Media/Pics/emot/train\sad\...
8	18	78	2	milk	D:/Media/Pics/emot/train\fear...

3) Количество положительных реакций по продуктам

```

SELECT COUNT(photos.id) as reactions, rea.positive_reactions,
(rea.positive_reactions*100/COUNT(photos.id))::numeric(6,2) as percent, photos.product as
product
from (
SELECT ph.id, ph.experiment, pr.name as product
from photolink ph
join experiment ex on (ph.experiment=ex.id)
join product pr on (ex.product_id=pr.id)
group by pr.id, ph.id, ex.id) as photos
join (SELECT COUNT(good_react.id) as positive_reactions, good_react.product as product
FROM (
SELECT ph.id, ph.experiment, pr.name as product, em.emot_name
from photolink ph
join experiment ex on (ph.experiment=ex.id)

```

```

join product pr on (ex.product_id=pr.id)
join emotion em on (em.id=emot_id)
WHERE (em.parent_id='11')
group by pr.id, ph.id, ex.id, em.emot_name)
as good_react group by good_react.product) as rea
on (rea.product=photos.product)
group by photos.product, rea.positive_reactions;

```

	reactions bigint	positive_reactions bigint	percent numeric (6,2)	product character varying (25)
1	2138	819	38.00	drink
2	1046	387	36.00	film
3	1064	409	38.00	syrok
4	355	62	17.00	milk
5	1088	397	36.00	cheese

4) Показывает возраст людей

```

select age_group, count(1) quantity from (
select
case
when p.age between 18 and 29 then '18-29'
when p.age between 30 and 39 then '30-39'
when p.age between 40 and 59 then '40-59'
when p.age between 60 and 80 then '60-80'
else null end age_group
from people p
) tmp
group by age_group order by age_group

```

	age_group text	quantity bigint
1	18-29	17
2	30-39	16
3	40-59	37
4	60-80	33

5) Распределение положительных реакций по полу

```

select em.emot_name, sum(case when pp.sex='1' then 1 end) as males,
sum(case when pp.sex='2' then 1 end) as females, pr.name as product
from emotion em
join photolink ph on ph.emot_id=em.id
join people pp on pp.id=ph.person
join experiment ex on ph.experiment=ex.id
join product pr on pr.id=ex.product_id
where em.parent_id is not null and ex.id=3

```

group by em.emot\_name, pr.name  
order by females desc;





	<div>emot_name</div> <div>character varying (25)</div>	<div>males</div> <div>bigint</div>	<div>females</div> <div>bigint</div>	<div>product</div> <div>character varying (25)</div>
1	happy	99	159	drink
2	neutral	67	98	drink
3	sad	62	91	drink
4	angry	53	87	drink
5	fear	69	63	drink
6	surprise	55	56	drink
7	disgust	13	7	drink

## 5. Физическое проектирование

### 5.1. Создание Базы Данных

```
1 SELECT * FROM public.emotion
2 ORDER BY id ASC
```

Data Output Explain Messages Notifications





	 id [PK] integer 	parent_id integer 	emot_name character varying (25) 
1	1	10	angry
2	2	10	disgust
3	3	10	fear
4	4	11	happy
5	5	8	neutral
6	6	8	sad
7	7	11	surprise
8	8	[null]	weak
9	9	[null]	strong
10	10	9	negative
11	11	9	positive

```

1 SELECT * FROM public.experiment
2 ORDER BY id ASC

```

Data Output Explain Messages Notifications





	 id [PK] integer 	product_id integer 	exp_date date 
1	1	1	2021-06-18
2	2	2	2021-06-18
3	3	2	2021-06-18
4	4	3	2021-06-07
5	5	4	2021-05-28
6	6	5	2021-06-12

```

1 SELECT * FROM public.people
2 ORDER BY id ASC

```







Data Output Explain Messages Notification

	 id [PK] integer 	age integer 	sex integer 
1	1	30	1
2	2	18	2
3	3	45	2
4	4	42	1
5	5	37	2
6	6	59	1
7	7	59	2
8	8	39	2
9	9	20	2
10	10	48	2
11	11	56	1
12	12	57	1
13	13	56	2
14	14	65	2
15	15	39	1

```

1 SELECT * FROM public.photolink
2 ORDER BY id ASC

```

Data Output		Explain	Messages	Notifications		
	id [PK] integer 	emot_id integer 	address character varying (1000) 	person integer 	experiment integer 	
1	13	1	D:/Media/Pics/emot/train\angr...	1	1	
2	14	1	D:/Media/Pics/emot/train\angr...	34	5	
3	15	1	D:/Media/Pics/emot/train\angr...	[null]	[null]	
4	16	1	D:/Media/Pics/emot/train\angr...	[null]	[null]	
5	17	1	D:/Media/Pics/emot/train\angr...	[null]	[null]	
6	18	1	D:/Media/Pics/emot/train\angr...	5	2	
7	19	1	D:/Media/Pics/emot/train\angr...	[null]	[null]	
8	20	1	D:/Media/Pics/emot/train\angr...	[null]	[null]	
9	21	1	D:/Media/Pics/emot/train\angr...	[null]	[null]	
10	22	1	D:/Media/Pics/emot/train\angr...	52	4	
11	23	1	D:/Media/Pics/emot/train\angr...	[null]	[null]	
12	24	1	D:/Media/Pics/emot/train\angr...	[null]	[null]	
13	25	1	D:/Media/Pics/emot/train\angr...	[null]	[null]	
14	26	1	D:/Media/Pics/emot/train\angr...	[null]	[null]	
15	27	1	D:/Media/Pics/emot/train\angr...	[null]	[null]	

```

1 SELECT * FROM public.product
2 ORDER BY id ASC

```

Data Output

Explain

Messages

Notifications

	<div>id</div> <div>[PK] integer</div>		<div>name</div> <div>character varying (25)</div>	
1		1	film	
2		2	drink	
3		3	syrok	
4		4	cheese	
5		5	milk	

## 5.2. Создание Таблиц

```
CREATE TABLE public.emotion
```

```
(
```

```
id integer NOT NULL DEFAULT nextval("Emotion_id_seq"::regclass),
```

```

parent_id integer,
emot_name character varying(25) COLLATE pg_catalog."default" NOT NULL,
CONSTRAINT "Emotion_pkey" PRIMARY KEY (id),
CONSTRAINT "Emotion_parent_id_fkey" FOREIGN KEY (parent_id)
    REFERENCES public.emotion (id) MATCH SIMPLE
    ON UPDATE NO ACTION
    ON DELETE NO ACTION
)
TABLESPACE pg_default;
ALTER TABLE public.emotion
    OWNER to postgres;

```

```

CREATE TABLE public.experiment
(
    id integer NOT NULL,
    product_id integer NOT NULL,
    exp_date date,
    CONSTRAINT experiment_pkey PRIMARY KEY (id)
)
TABLESPACE pg_default;
ALTER TABLE public.experiment
    OWNER to postgres;

```

```

CREATE TABLE public.people
(
    id integer NOT NULL,
    age integer,
    sex integer,
    CONSTRAINT "People_pkey" PRIMARY KEY (id)
)
TABLESPACE pg_default;
ALTER TABLE public.people
    OWNER to postgres;

```

```

CREATE TABLE public.photolink
(
    id integer NOT NULL DEFAULT nextval("PhotoLink_id_seq"::regclass),
    emot_id integer NOT NULL,
    address character varying(1000) COLLATE pg_catalog."default" NOT NULL,
    person integer,
    experiment integer,
    CONSTRAINT "PhotoLink_pkey" PRIMARY KEY (id),
    CONSTRAINT "PhotoLink_emot_id_fkey" FOREIGN KEY (emot_id)
        REFERENCES public.emotion (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION,
    CONSTRAINT photolink_experiment_fkey FOREIGN KEY (experiment)
        REFERENCES public.experiment (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT photolink_person_fkey FOREIGN KEY (person)
        REFERENCES public.people (id) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
)
TABLESPACE pg_default;

ALTER TABLE public.photolink
    OWNER to postgres;

CREATE TABLE public.product
(
    id integer NOT NULL,
    name character varying(25) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT product_pkey PRIMARY KEY (id)
)
TABLESPACE pg_default;

```



ALTER TABLE public.product  
OWNER to postgres;

### 5.3. Заполнение таблиц

Приводятся скриншоты с содержимым таблиц (в пункте 5.1).

### 5.4. Запросы в терминах SQL

Приводятся SQL-скрипты запросов, сформулированных в п. 1.7, и результаты их выполнения (в пункте 4.4).

### 5.5. Оценка размеров Базы Данных и каждого из файлов

Необходимо заполнить таблицу следующего вида.

Отношение	Атрибут	Тип данных	Размер, байт	Среднее количество	Объем, байт
Photolink	id	INTEGER	8	10000	40 280 000
	address	VARCHAR(1000)	4000		
	experiment	INTEGER	4		
	person	INTEGER	8		
	experiment	INTEGER	4		
	emot_id	INTEGER	4		
Emotion	id	INTEGER	4	10	1080
	parent_id	INTEGER	4		
	name	CHARACTER VARING (25)	100		
Experiment	id	INTEGER	4	20	240
	product_id	INTEGER	4		
	exp_date	DATE	4		
People	id	INTEGER	4	1000	12 000
	age	INTEGER	4		
	sex	INTEGER	4		
Product	id	INTEGER	4	50	5 200

	name	CHARACTER VARIING (25)	100		
--	------	---------------------------	-----	--	--

Средний размер базы данных - 38,4317 мегабайт.

## 5.6 Заполнение базы данных

Часть кода для заполнения базы данными на языке python.

```
p = os.path.join(const_train_path)
os.chdir(p)
print(os.getcwd())
menu = os.listdir(p)
print(menu)
count = len(menu)
sqlEmo = ("INSERT INTO emotion (id, emot_name) values ('%s', '%s');")
sqlPhoto = ("INSERT INTO photolink (emot_id, address) values ('%s', '%s');")
for i in range(count):
    cursor.execute(sqlEmo % (i+1, menu[i]))
    p = os.path.join(const_train_path, menu[i])
    PicNames = os.listdir(p)
    print(menu[i], len(PicNames))
    for PicName in PicNames:
        address = os.path.join(p, PicName)
        #print(address)
        cursor.execute(sqlPhoto % (i+1, address))

while j!= 3+1:
#выбор папки эмоции
    str=p.split("\\")
    cur = str[-1]
    curnum = indexset(cur)
    address = p + '\\' + os.listdir(p)[j]
    print(address)
```

```

state = 'INSERT INTO photolink (emot_id, address) values ( :id , :line);'
print(state)
#state = composition(curnum, address)
try:
    print(state, {'id':curnum, 'line': address })
    cursor.execute(state, {'id':curnum, 'line': address })
except:
    print('meh')
j= j+1
#КОНЕЦ ВСТАВКИ

```

```

p = os.path.join('D:\Media\Pics\emot\\train\\angry')
os.chdir(p)
print(os.getcwd())
#print(os.listdir(p))
#функции на каждый сложный процесс
def composition(index, line):
    compose = ("INSERT INTO photolink (emot_id, address) values (", str(index), " , '", line, "');"
    statement = "
    statement = state.join (compose)
    return statement

```

```

def indexset(catalog):

```

```

    options ={
        'angry': 1,
        'happy': 2,
        'neutral': 3,
        'sad': 4,
        'suprise': 5
    }

```

```

index = options[catalog]

```

