
СОФИЙСКИ УНИВЕРСИТЕТ
„СВ. КЛИМЕНТ ОХРИДСКИ“

ФАКУЛТЕТ ПО
МАТЕМАТИКА И ИНФОРМАТИКА



SOFIA UNIVERSITY
ST. KLIMENT OHRIDSKI

FACULTY OF
MATHEMATICS AND INFORMATICS

Проект по Бази от данни

Тема:
Museum Database Manager



Изготвено от:
Калоян Николов
Светлин Попиванов
Андрей Стоев
Таня Желева
Десислава Желязкова

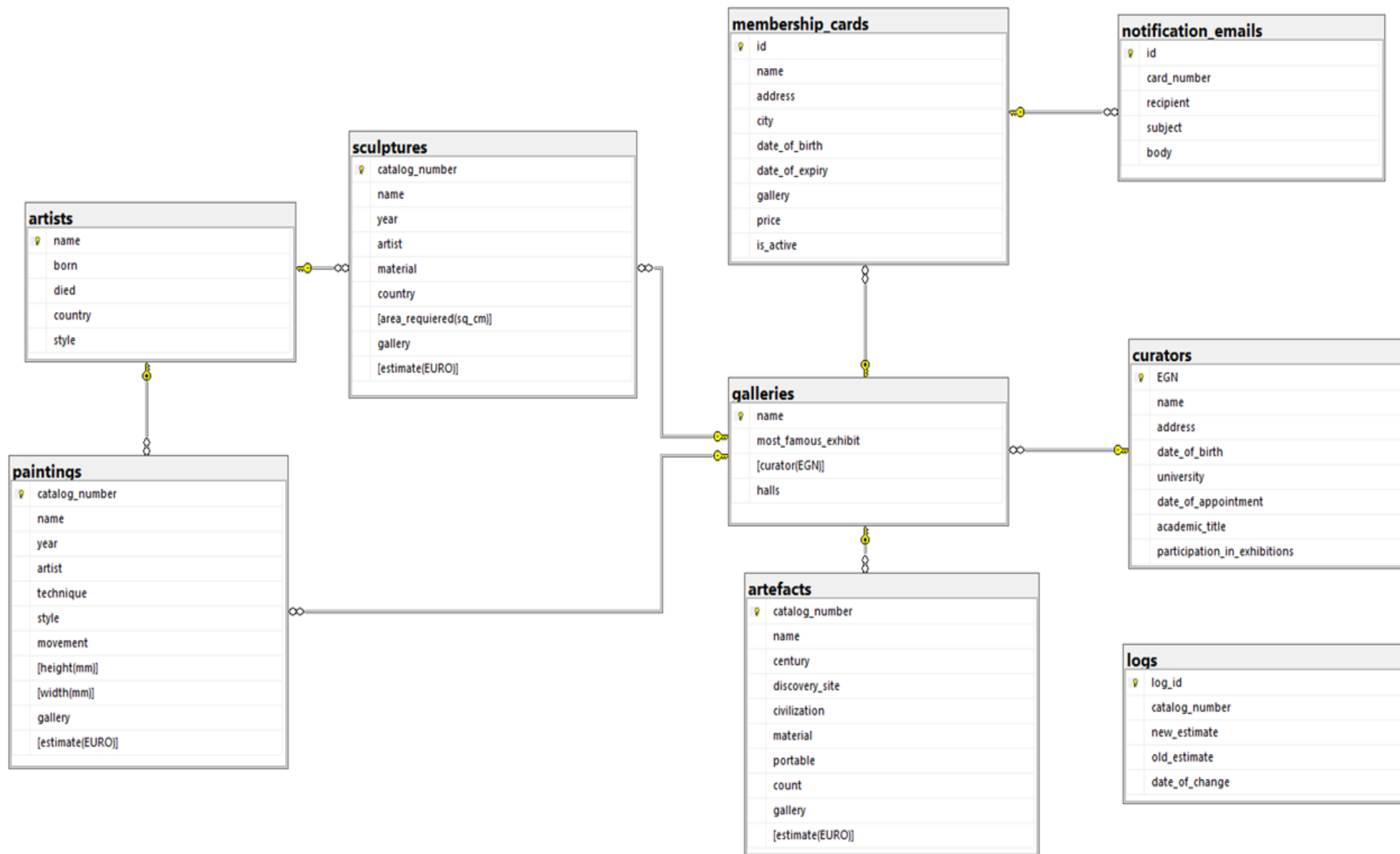
Тема на проекта:

Базата от данни е предназначена за музей с изложби на картини, артефакти и скулптури от различни периоди на човешката история. Съдържа се детайлна информация за всички налични експонати, галериите в които са разположени, техните куратори и броят зали на всяка от галериите. Съхраняват се данни за всички клиенти, притежаващи активни или изтекли посетителски карти, както и кратка информация за тях. Базата от данни разполага с история, касаеща всички промени в цените на експонатите и по този начин следи тенденциите в съответните направления на изкуството. Базата от данни поддържа информация за изпращане на имейли на клиентите с изтекли посетителски карти. Тази таблица лесно може да се използва от приложението, което се асоциира с музея.

Базата от данни, която е разработена, е полезна за разрешаване на редица проблеми, свързани не само с организацията на музея, но и с услугите, които предлага интерфейса на приложението, което е част от маркетинга на музея:

- Базата от данни е полезна за потребители, които не са посещавали никога музея и искат да проверят какви експонати са налични от даден тип/сфера или друг параметър от базата данни;
- В случай на изчезване на експонати поради кражба или увреждане от друго естество, базата от данни ще помогне при определянето на липсващите или увредени предмети и оценка на тяхната стойност;
- Лесно конфигуриране на гостуващи изложби – предназначени за изложения в чужди музеи и държави, на базата на тенденциите в даден момент или друг вид неинституционална изложба;
- Базата от данни ще бъде незаменим помощник на изследователите, отдали живота си на творчеството от дадена епоха, стил или конкретен творец;
- Съхранените данни могат да служат за статистически извадки за приходите от посетителски карти, тяхната стойност и колко от тях са активни във всеки един момент. Това е в пряка помощ на бизнес логиката на музейното приложение, както и на музея като бизнес организация;
- Базата от данни ще бъде от помощ за всички ученици и студенти, изучаващи произведения на творци, включени в музея;

1. Схема на релациите



Таблицы, участващи в базата данни:

1.1. Картини (каталоген номер, име, година на създаване, автор, техника, стил, творчески период, ширина, дължина, изложбена галерия, оценъчна стойност)

```
CREATE TABLE paintings(  
    catalog_number INT IDENTITY(10000,1) NOT NULL,  
    name VARCHAR(100) NOT NULL,  
    year SMALLINT,  
    artist VARCHAR(50) NOT NULL,  
    technique CHAR(20) NOT NULL,  
    style CHAR(20) NOT NULL,  
    movement CHAR(30) NOT NULL,  
    [height(mm)] SMALLINT NOT NULL,  
    [width(mm)] SMALLINT NOT NULL,  
    gallery CHAR(30) NOT NULL,  
    [estimate(EURO)] INT NOT NULL,  
);
```

1.2. Артефакти (каталоген номер, име, датировка, място на откриване, цивилизация, материал, преносим, брой съставни части, галерия, оценъчна стойност)

```
CREATE TABLE artefacts(  
    catalog_number INT IDENTITY(40000,1) NOT NULL,  
    name VARCHAR(50) NOT NULL,  
    century CHAR(15),  
    discovery_site VARCHAR(30),  
    civilization CHAR(20),  
    material CHAR(20) NOT NULL,  
    portable CHAR(1) NOT NULL,  
    count SMALLINT NOT NULL,  
    gallery CHAR(30) NOT NULL,  
    [estimate(EURO)] INT NOT NULL  
);
```

1.3. Скулптури (каталоген номер, име, година на създаване, автор, материал, държава на произход, нужна площ, галерия, оценъчна стойност)

```
CREATE TABLE sculptures(  
    catalog_number INT IDENTITY(70000,1) NOT NULL,  
    name VARCHAR(50) NOT NULL,  
    year SMALLINT,  
    artist VARCHAR(50),  
    material CHAR(20) NOT NULL,  
    country CHAR(20) NOT NULL,  
    [area_required(sq_cm)] SMALLINT NOT NULL,  
    gallery CHAR(30) NOT NULL,  
    [estimate(EURO)] INT NOT NULL  
);
```

1.4 Автори (име, година на раждане, година на смърт, държава, творчески стил)

```
CREATE TABLE artists(  
    name VARCHAR(50) NOT NULL,  
    year SMALLINT,  
    country CHAR(20) NOT NULL,  
    style CHAR(20) NOT NULL,  
    [period_start] SMALLINT NOT NULL,  
    [period_end] SMALLINT NOT NULL,  
    [gallery] CHAR(30) NOT NULL,  
    [estimate(EURO)] INT NOT NULL  
);
```

```
name VARCHAR(50) NOT NULL,  
born SMALLINT,  
died SMALLINT,  
country CHAR(20) NOT NULL,  
style CHAR(30) NOT NULL,  
CONSTRAINT uk_artists UNIQUE (name)  
);
```

1.5. Посетителски карти (уникален сериен номер, име на притежател, адрес, град, рождена дата, дата на изтичане, издадена за галерия, цена, активна(да/не))

```
CREATE TABLE membership_cards(  
id INT IDENTITY(30000, 1) NOT NULL,  
name VARCHAR(30) NOT NULL,  
address VARCHAR(80) NOT NULL,  
city VARCHAR(30) NOT NULL,  
date_of_birth DATE NOT NULL,  
date_of_expiry DATE NOT NULL,  
gallery CHAR(30) NOT NULL,  
price DECIMAL(4,2) NOT NULL  
);
```

* колона “активна(да/не)” се добавя преди създаването на съответния тригер, тъй като тя ще служи за гарантирането на soft delete и ще се ползва от него.

1.6. Галерии (име, най-забележителен експонат, куратор, брой зали)

```
CREATE TABLE galleries(  
name CHAR(30) NOT NULL,  
most_famous_exhibit VARCHAR(50) NOT NULL,  
[curator(EGN)] CHAR(10) NOT NULL,  
halls SMALLINT NOT NULL,  
CONSTRAINT uk_galleries UNIQUE (name)  
);
```

1.7. Куратори (ЕГН, име, адрес, дата на раждане, университет, дата на назначаване, научна степен, брой участия в изложби)

```
CREATE TABLE curators(  
EGN CHAR(10) NOT NULL,  
name CHAR(30) NOT NULL,  
address VARCHAR(80) NOT NULL,  
date_of_birth DATE NOT NULL,  
university VARCHAR(50),  
date_of_appointment DATE NOT NULL,  
academic_title CHAR(30) NOT NULL,  
participation_in_exhibitions SMALLINT NOT NULL,  
CONSTRAINT uk_curators UNIQUE (EGN)  
);
```

1.8. Лог на оценъчните цени (уникален пореден номер, каталожен номер на продукта, нова цена, стара цена, дата на промяна)

```
CREATE TABLE logs(
    log_id INT IDENTITY(1,1) PRIMARY KEY,
    catalog_number INT,
    new_estimate INT,
    old_estimate INT,
    date_of_change DATE
);
```

1.9. Email лог (уникален пореден номер, номер на карта, получател, тема, съдържание)

```
CREATE TABLE notification_emails(
    id INT PRIMARY KEY IDENTITY(1,1),
    card_number INT,
    recipient VARCHAR(30),
    subject VARCHAR(80),
    body VARCHAR(150)
);
```

3. ПРИМЕРИ С ПРОСТИ ЗАЯВКИ И ЗАЯВКИ ВЪРХУ ДВЕ ИЛИ ПОВЕЧЕ РЕЛАЦИИ

3.1. Да се намерят имената на всички картини, чиято стойност е повече от 11000 евро.
Заявка:

```
SELECT paintings.name
FROM paintings
where [estimate(EURO)] > 11000
GO
```

Резултат:

	name		
1	The Tower of the Palazzo del Podesta		
2	Matin d'avril		
3	A Game of Croquet	20	Mongyu dowondo (Dream Journey to...
4	The Play Scene in Hamlet	21	Tibetan girls eyes
5	Jungle	22	Soul is like wind; Life is like Snow
6	View of an Italian Hill Town	23	Landscape with an Eye
7	Westminster	24	Flowers and Butterflies
8	A street scene with trees in summer	25	A French Prisoner-of-War
9	Still Life - Leaf, Shell, and Jacks	26	Ainsworth
10	The Red Rook	27	Moscow in Winter from the Sparrow ...
11	Tulip	28	Jesus walking on water
12	The Last Emperor	29	The portrait of Helena Modrzejewska
13	Anna van Hannover, princess of Ora...	30	Military exercises of 1885
14	Clearing Up-Coast of Sicily	31	Nativity of the Virgin
15	Flower Seller	32	Annunciation
16	Surf at Hvide Sande	33	Annunciation
17	Bandits Shooting at Travellers	34	Scenery
18	River landscape	35	Solidarity
19	????? - Eight views of the Four Seas...	36	The Ninth Wave

3.2. Да се намерят цените на всички абонаментни карти за галерията 'Ancient Greece'.

Заявка:

```
SELECT price
FROM membership_cards
WHERE gallery='Ancient Greece'
GO
```

Резултат:

	price		
1	49.68		
2	78.65		
3	54.29	14	63.94
4	61.76	15	57.02
5	54.41	16	21.19
6	29.58	17	57.14
7	41.02	18	97.02
8	86.98	19	69.96
9	55.85	20	57.41
10	50.48	21	70.00
11	21.67	22	38.45
12	20.01	23	21.99
13	88.42	24	57.13

3.3. Да се намерят имената на всички галерии, където има артефакти от Древен Египет, без да се дублират.

Заявка:

```
SELECT DISTINCT gallery
FROM artefacts
WHERE artefacts.civilization='Ancient Egypt'
GO
```

Резултат:

	gallery
1	Ancient Egypt

3.4. Да се намерят имената и адресите на всички хора с абонаментни карти, чиито адреси започват с '2'.

Заявка:

```
SELECT name,address
FROM membership_cards
WHERE address LIKE '2 %'
GO
```

Резултат:

	name	address
1	Katherine Lowe	2 Kingswood Estate
2	Mysha Haworth	2 Basildon Court, Croxley Road
3	Lewys Odonnell	2 Windemere
4	Roisin Scott	2 Rostheme Avenue
5	Maha Wilkes	2 Kings Drive
6	Saskia Lewis	2 Judge Walk
7	Mylie Mathews	2 Drift Cottages
8	Hettie Russo	2 Salter Close
9	Lilly-Rose Cote	2 Henry Street
10	Aleesha Patton	2 The Boume
11	Ayyub Rowe	2 Lewisham Street
12	Aisha Bennett	2 Rabbits Road
13	Ella-May Whit...	2 Old Common Road
14	Rhonda Buxton	2 The Avenue

3.5. Да се намерят имената и датите на започване на работа на всички куратори, които са завършили Yale University. Задайте псевдоними на атрибутите за за име и датата съответно 'name' и 'Started working on'.

Заявка:

```
SELECT name AS 'Curator', date_of_appointment AS 'Started working on'
FROM curators
WHERE university = 'Yale University'
GO
```

Резултат:

	Curator	Started working on
1	Troy Ferguson	2003-06-22
2	Carmen White	2006-01-09
3	George Rivera	2006-11-13
4	Adrian Moha...	2004-06-02
5	Layla Cook	2010-12-04

3.6. Да се намерят имената на всички художници, които са от Италия и стилът на картините им е 'Landscape'.

Заявка:

```
(select artist from paintings where style='Landscape')
INTERSECT
(select name from artists where country='Italy')
GO
```

Резултат:

	artist
1	Giuseppe Abbati

3.7. Да се намерят всички картини и скулптури, чиито цени са поне 500 000 евро.

Заявка:


```
(select distinct paintings.name,paintings.[estimate(EURO)]
from paintings
where paintings.[estimate(EURO)]>=500000 )
UNION
(select distinct sculptures.name,sculptures.[estimate(EURO)]
from sculptures
where sculptures.[estimate(EURO)]>=500000)
GO
```

Резултат:

	name	estimate(EURO)
1	Alfarera de Raquira	842000
2	Baruch	600000
3	Isaiah	600000
4	Maurizio Frangipa...	600000
5	Nierozpoznani ("...	600000
6	Project for the Mo...	750000
7	The Seine and th...	652000
8	The Triumph of N...	600000

3.8. Да се намерят имената на галериите, в които има артефакти от камък, без тези, в които има картини, чиито имена започват с 'A'.

Заявка:

```
(select gallery from artefacts where artefacts.material='stone')
EXCEPT
(select gallery from paintings where paintings.name LIKE 'A%' )
GO
```

Резултат:

	gallery
1	Ancient Egypt
2	Ancient Greece
3	Asian
4	Rome

3.9. Да се намерят имената на картините, които са нарисувани след 2000 година или са най-известното изложение в галерията. Задайте псевдоним на атрибута за име на картината - 'painting'.

Заявка:

```
select DISTINCT paintings.name as painting
from paintings, galleries
where paintings.name = galleries.most_famous_exhibit
OR paintings.year>2000
GO
```

Резултат:

	painting
1	Flowers and Butterflies
2	Study Of A Stone
3	The Last Emperor
4	View from the Wine ...
5	Westminster
6	Wild Geese
7	Yellow I

3.10. Да се намерят имената на галериите и кураторите, които са завършили Stanford University. Задайте псевдоним на атрибутите за име съответно gallery и curator.

Заявка:

```
select DISTINCT galleries.name as gallery, curators.name AS curator
from galleries,curators
where curators.university='Stanford University'
AND galleries.[curator(EGN)]=curators.EGN
```

GO

Резултат:

	gallery	curator
1	Ancient Greece	Paula Jenkins
2	Asian	Lois Armstrong
3	Expressionism	Philippa Sanders
4	Renaissance	Kristina Arnold

4. ПОДЗАЯВКИ

4.1. Намерете онези артефакти, които са по-скъпи от най-известните такива в галериите.

Заявка:

```
SELECT a.name AS artefact_name, a.gallery AS gallery_name, a.[estimate(EURO)] AS [price(EURO)]
FROM artefacts a
WHERE a.[estimate(EURO)] > ALL (SELECT a.[estimate(EURO)]
FROM galleries g
LEFT JOIN artefacts a ON a.name = g.most_famous_exhibit
WHERE a.name IS NOT NULL)

ORDER BY a.name;
GO
```

Резултат:

	artefact_name	gallery_name	price(EURO)
1	Ivory pomegranate	Ancient Egypt	956000
2	Nefertiti Bust	Ancient Egypt	1000000

4.2. Намерете всички хора, които имат възможността да присъстват на изложенията на най-опитните (най-много участия в изложби) куратори.

Заявка:

```

SELECT m.name AS curator_name
FROM membership_cards m
WHERE m.gallery IN (SELECT g.name
                    FROM galleries g
                    WHERE g.[curator(EGN)] IN (SELECT c.egn
                                              FROM curators c
                                              WHERE c.participation_in_exhibitions >= ALL (SELECT participation_in_exhibitions
                                                                                          FROM curators)))

ORDER BY m.name;
GO

```

Резултат:

	curator_name		
1	Abby Finley		
2	Aleesha Patton	23	Jacqueline Vale...
3	Alexandra Church	24	Jacques Higgins
4	Amritpal Casey	25	Kajus Whitworth
5	Ava-May Paul	26	Killian Woodcock
6	Ayyub Rowe	27	Lily-Mai Oneal
7	Bethan Sherman	28	Mabel Obrien
8	Betty Kline	29	Maha Wilkes
9	Billie Gross	30	Malik Lyon
10	Brax Guerra	31	Marshall Rubio
11	Caoimhe Pratt	32	Menaal Gibson
12	Caspian Santana	33	Myron Petty
13	Cecil Callaghan	34	Nusaybah Mair
14	Craig Dodd	35	Rahim Buxton
15	Drake Mcculloch	36	Rhonda Buxton
16	Eamonn Barton	37	Riya Wormald
17	Faraz Knox	38	Skylar Bates
18	Florrie English	39	Tasnim Phelps
19	Harlan Naylor	40	Tyrone Cox
20	Hira Stuart	41	Waseem Strickl...
21	Horace Winters	42	Wil Rooney
22	Inara Norris	43	Zaydan Greaves

4.3. За всеки човек, който се казва Scott Norton, намерете онези художници/скулптористи, чиито творби може да разгледа на намалена цена.

Заявка:

```

SELECT p.artist AS artist_name, p.gallery AS gallery_name, t1.name, t1.address, t1.city, t1.date_of_birth
FROM paintings p
JOIN (SELECT gallery, name, address, city, date_of_birth
      FROM membership_cards
      WHERE name LIKE 'Scott Norton') t1 ON t1.gallery = p.gallery
UNION
SELECT s.artist AS artist_name, s.gallery AS gallery_name, t2.name, t2.address, t2.city, t2.date_of_birth
FROM sculptures s
JOIN (SELECT gallery, name, address, city, date_of_birth
      FROM membership_cards
      WHERE name LIKE 'Scott Norton') t2 ON t2.gallery = s.gallery
ORDER BY gallery_name;
GO

```

Резултат:

	artist_name	gallery_name	name	address	city	date_of_birth
1	Aleijadinho(Antonio Francisco Lisboa)	Abstract Expressionism	Scott Norton	21 Brook Street	Manningtree	1977-10-03
2	Diana Al-Hadid	Abstract Expressionism	Scott Norton	21 Brook Street	Manningtree	1977-10-03
3	Evelyn Abelson	Abstract Expressionism	Scott Norton	21 Brook Street	Manningtree	1977-10-03
4	Gertrude Abercrombie	Abstract Expressionism	Scott Norton	21 Brook Street	Manningtree	1977-10-03
5	Ismail Acar	Abstract Expressionism	Scott Norton	21 Brook Street	Manningtree	1977-10-03
6	Josef Abel	Abstract Expressionism	Scott Norton	21 Brook Street	Manningtree	1977-10-03
7	Knud Agger	Abstract Expressionism	Scott Norton	21 Brook Street	Manningtree	1977-10-03
8	Per Abramsen	Abstract Expressionism	Scott Norton	21 Brook Street	Manningtree	1977-10-03
9	Rowena Meeks Abdy	Abstract Expressionism	Scott Norton	21 Brook Street	Manningtree	1977-10-03
10	Abed Abdi	Futurism	Scott Norton	18 Prospect S...	Liverpool	1961-02-18
11	Alessandro Algardi	Futurism	Scott Norton	18 Prospect S...	Liverpool	1961-02-18
12	Andreas Achenbach	Futurism	Scott Norton	18 Prospect S...	Liverpool	1961-02-18
13	Edgar Ainsworth	Futurism	Scott Norton	18 Prospect S...	Liverpool	1961-02-18
14	Francesco Albani	Futurism	Scott Norton	18 Prospect S...	Liverpool	1961-02-18
15	József Ács	Futurism	Scott Norton	18 Prospect S...	Liverpool	1961-02-18
16	Julio Abril	Futurism	Scott Norton	18 Prospect S...	Liverpool	1961-02-18
17	Per Abramsen	Futurism	Scott Norton	18 Prospect S...	Liverpool	1961-02-18

4.4. Изведете името и адреса на куратора, който поддържа галерията, в която се намира най-скъпия и известен експонат.

Заявка:

```

SELECT c.name AS curator_name, c.egn
FROM curators c
WHERE c.egn LIKE (SELECT TOP 1 g.[curator(EGN)]
                  FROM galleries g
                  JOIN (SELECT p.name, p.[estimate(EURO)] AS price
                        FROM paintings p
                        UNION
                        SELECT a.name, a.[estimate(EURO)] AS price
                        FROM artefacts a
                        UNION
                        SELECT s.name, s.[estimate(EURO)] AS price
                        FROM sculptures s) t ON t.name = g.most_famous_exhibit
                  ORDER BY t.price DESC);
GO

```

Резултат:

	curator_name	egn
1	Paula Jenkins	9352544582

4.5. Да се изведе името и цената на най-скъпия експонат, който се намира в тази галерия, чиято посетителска карта е най-скъпа.

Заявка:

```
SELECT TOP 1 t2.name AS exhibit_name, t2.price, g.name AS gallery_name
FROM galleries g
JOIN (SELECT m.gallery AS gallery_name
      FROM membership_cards m
      WHERE m.price >= ALL (SELECT m1.price
                           FROM membership_cards m1)) t1 ON g.name = t1.gallery_name
JOIN (SELECT p.gallery, p.name, p.[estimate(EURO)] AS price
      FROM paintings p
      UNION
      SELECT a.gallery, a.name, a.[estimate(EURO)] AS price
      FROM artefacts a
      UNION
      SELECT s.gallery, s.name, s.[estimate(EURO)] AS price
      FROM sculptures s) t2 ON t2.gallery = g.name
ORDER BY t2.price DESC;
GO
```

Резултат:

	exhibit_name	price	gallery_name
1	Nefertiti Bust	1000000	Ancient Egypt

5. СЪЕДИНЕНИЯ

5.1. Изведете имената, адреса и града на местообитание на хората, които имат намаление на посещаването на галерии, които имат скулптори от Италия.

Заявка:

```
SELECT DISTINCT m.name, m.address, m.city
FROM membership_cards m
JOIN galleries g ON g.name = m.gallery
JOIN sculptures s ON s.gallery = g.name
WHERE s.country LIKE 'Italy'
ORDER BY m.name;
GO
```

Резултат:

	name	address	city
1	Abdul Chapman	55 Snipefield Lane	Shere
2	Abigail Cain	25 Cone Lane	Silkstone Common
3	Aleesha Squires	74 Heathpark Drive	Windlesham
4	Andre Manning	153 Breckfield Road North	Liverpool
5	Ariella Henson	Far Leys	Southwell
6	Aurelia Crane	25 Middlebrook Green	Market Harborough
7	Chantel Alston	24 Bathurst Road	Gloucester
8	Dionne Glass	150 Wells Road	Glastonbury
9	Donte Bowden	11 Lyndhurst	Middle Drove
10	Elana John	Hamston Fawr	Dyffryn
11	Ema Bell	44 Tors Park	Lynmouth
12	Jae Gamble	1 Bollinder Place	Islington
13	Karson Christian	27 Parkfield Crescent	Wolverhampton
14	Lylah Ridley	7 Mill Weir Gardens	Liverpool
15	Missy Mccarthy	13 Trelyn	Shocklach
16	Mylie Mathews	2 Drift Cottages	Rossett
17	Pierce Woodw...	6 The Crestway	Whetstone
18	Roisin Scott	2 Rostherne Avenue	Great Sutton
19	Rosa Rawlings	78 Wordsworth Drive	Sutton
20	Scott Norton	18 Prospect Street	Liverpool
21	Sebastian Ma...	30 Canterbury Road	Beaufort
22	Shanna Drew	20 Westfield	Bradninch
23	Sumaiya Wick...	3 Colne Road	Tonbridge

5.2. Изведете цената на онези карти за галерии, които са били продадени поне 2 пъти.

Заявка:

```
SELECT DISTINCT m1.price
FROM membership_cards m1
CROSS JOIN membership_cards m2
WHERE m2.name != m1.name AND m1.price = m2.price
ORDER BY m1.price;
GO
```

Резултат:

	price
1	52.36
2	52.85
3	53.91
4	58.31
5	63.78
6	65.81
7	68.00
8	71.83
9	98.35

5.3. Намерете онези хора, които имат карти за безплатен вход в галерии, в които кураторите са по-млади от тях и има артефакти, датиращи от преди Христа.

Заявка:

```
SELECT DISTINCT m.name, m.date_of_birth
FROM membership_cards m
JOIN curators c ON c.date_of_birth < m.date_of_birth
JOIN galleries g ON g.[curator(EGN)] = c.egn AND g.name = m.gallery
JOIN artefacts a ON a.gallery = g.name
WHERE a.century LIKE '%BC'
ORDER BY m.name;
GO
```

Резултат:

	name	date_of_birth			
1	Abubakar Knight	1982-10-07			
2	Alanah Case	1951-01-19	27	Ilayda Bentley	1994-11-13
3	Alastair Stone	1979-10-26	28	Jana Curran	1974-12-14
4	Alec Pemberton	1949-05-16	29	Judith Dunlop	1970-02-13
5	Angelika Terry	1977-03-11	30	Juniper Grant	1993-01-04
6	Anne Sloan	1973-08-04	31	Kade Summers	1957-06-16
7	Aqib Mosley	1975-12-04	32	Kadie Rios	1995-02-22
8	Artur Keeling	1969-01-08	33	Kierran Bevan	1974-09-10
9	Cherise Walters	1955-04-30	34	Kurt Carney	1979-10-09
10	Cleveland Price	1988-06-11	35	Lexi-May Mckee	1978-04-22
11	Dana Bernard	1978-11-30	36	Lilly-Rose Cote	1974-05-15
12	Dannielle Blake	1979-07-04	37	Lynn Ramos	1973-07-18
13	Dixie Sparks	1978-01-09	38	Manav Mckenna	1992-07-16
14	Duke Nguyen	1960-02-16	39	Neha Fitzpatrick	1980-10-11
15	Eddison Brady	1999-06-20	40	Poppy Howells	1986-06-22
16	Elodie Deacon	1976-04-27	41	Rahim James	1975-08-16
17	Fahima Koch	1997-09-14	42	Rajan Mckeown	1966-05-22
18	Felicity Parrish	1946-09-25	43	Rukhsar Howard	1976-09-14
19	Franklyn Charl...	1984-06-02	44	Samson Mcdo...	1958-03-21
20	Frederic Suarez	1945-10-11	45	Stefan Mills	1990-11-14
21	Gerrard Crawf...	1968-01-31	46	Steven Stanley	1981-09-03
22	Habibah Cantu	1978-01-11	47	Teo Larsen	2002-12-03
23	Hajrah Oconnell	2003-04-15	48	Tierney Gordon	1967-07-06
24	Harmony Coch...	1984-12-14	49	Traci Talley	1971-01-13
25	Heather Hodge	1991-07-14	50	Ty Fischer	1972-11-26
26	Herbert Diaz	1988-03-10			
51	Vivaan May	1999-11-09			
52	Wilson Blundell	1997-06-03			
53	Wren Black	2001-05-25			
54	Yvonne Person	1982-09-07			
55	Zander Weston	1997-07-04			

5.4. Изведете имената на онези куратори, които имат докторска степен и поддържат галерии, в които има преносими артефакти.

Заявка:

```

SELECT DISTINCT c.name AS curator_name, c.academic_title, g.name AS gallery_name
FROM galleries g
JOIN artefacts a ON a.gallery = g.name
RIGHT JOIN curators c ON c.egn = g.[curator(EGN)]
WHERE a.portable LIKE 'y' AND c.academic_title LIKE 'Doctor%';
GO

```

Резултат:

	curator_name	academic_title	gallery_name
1	Adrian Mohamed	Doctor of Philosophy	Ancient Egypt
2	Lois Armstrong	Doctor of Philosophy	Asian

5.5. Изведете имената и ЕГН-тата на онези куратори, които поддържат галерии, в които има само артефакти.

Заявка:

```

SELECT DISTINCT c.name, c.egn
FROM galleries g
LEFT JOIN paintings p ON p.gallery = g.name
LEFT JOIN sculptures s ON s.gallery = g.name
LEFT JOIN artefacts a ON a.gallery = g.name
LEFT JOIN curators c ON c.egn = g.[curator(EGN)]
WHERE p.name IS NULL AND s.name IS NULL;
GO

```

Резултат:

	name	egn
1	Adrian Mohamed	7328739042
2	Paula Jenkins	9352544582

6. ПРИМЕРИ С ГРУПИРАНЕ И АГРЕГАЦИЯ

6.1. Да се изведе за всяка от галериите броя на тези автори на скулптури, които имат повече от 1 скулптура, изложена в музея и скулптурите им се оценяват средно на поне 100000 евро.

Заявка:

```

SELECT s1.gallery, COUNT(DISTINCT artist) AS number_of_artists
FROM sculptures s1
WHERE s1.artist IN (SELECT artist
                    FROM sculptures
                    GROUP BY artist
                    HAVING COUNT(*) > 1 AND AVG([estimate(EURO)]) > 100000)
GROUP BY s1.gallery
ORDER BY number_of_artists DESC
GO

```

Резултат:

	gallery	number_of_artists
1	Futurism	4
2	Expressionism	3
3	Gothic	2
4	Abstract Expressionism	2
5	Asian	2
6	Classicism	2
7	Renaissance	1

6.2. Да се изведат имената на галериите с най-ниска и най-висока средна цена на посетителска карта, броя на картите за тях и средната цена на картите за тези галерии.

Заявка:

```

SELECT gallery, COUNT(*) AS cards, CAST(AVG(price) AS DECIMAL(4,2)) AS avg_price
FROM membership_cards m
GROUP BY gallery
HAVING AVG(m.price) <= ALL(SELECT AVG(price)
                        FROM membership_cards
                        GROUP BY gallery)
OR AVG(m.price) >= ALL(SELECT AVG(price)
                      FROM membership_cards
                      GROUP BY gallery)
GO

```

Резултат:

	gallery	cards	avg_price
1	Abstract Expressionism	22	64.02
2	Expressionism	22	53.76

6.3. За всички автори на скулптури, които творят към днешна дата, да се изведат име, брой техни скулптури, изложени в музея, стойността на най-скъпата им творба и средната стойност на всички техни скулптури.

Заявка:

```

SELECT artist, COUNT(*) AS number_of_sculptures, MAX([estimate(EURO)]) AS most_expensive_sculptures,
              AVG([estimate(EURO)]) AS avg_price
FROM sculptures
WHERE artist in (SELECT name
                FROM artists
                WHERE died IS NULL)
GROUP BY artist
GO

```

Резултат:

	artist	number_of_sculptures	most_expensive_sculptures	avg_price
1	Amber Aguirre	1	110100	110100
2	Diana Al-Hadid	2	110100	65050
3	Kinji Akagawa	4	150300	130200
4	Yaacov Agam	2	110100	65050

6.4. Да се изведе преоценка на стойността на всички картини, скулптури и артефакти от галерия "Класицизъм" в US долари. Приемаме, че текущият курс на евро-долар е 1,09. Резултатът да се закръгли до цяло число надолу.

Заявка:

```
SELECT FLOOR((SUM(ISNULL(p.[estimate(EURO)],0)) +
SUM(ISNULL(s.[estimate(EURO)],0)) + SUM(ISNULL(a.[estimate(EURO)],0))) * 1.09) AS [estimate(USD)]
FROM paintings p
FULL JOIN sculptures s ON p.catalog_number = s.catalog_number
FULL JOIN artefacts a ON p.catalog_number = a.catalog_number
WHERE p.gallery LIKE 'Classicism' OR s.gallery LIKE 'Classicism' OR a.gallery LIKE 'Classicism'
GO
```

Резултат:

	estimate(USD)
1	1380594

6.5. Да се изведе общият брой експонати във всяка галерия. Резултатът да се сортира в низходящ ред спрямо броя на експонатите в галерията.

Заявка:

```
SELECT t.gallery_name, COUNT(*) AS exhibit_count
FROM (SELECT g.name AS gallery_name, p.name AS exhibit_name
FROM galleries g
JOIN paintings p ON p.gallery = g.name
UNION
SELECT g.name AS gallery_name, a.name AS exhibit_name
FROM galleries g
JOIN artefacts a ON a.gallery = g.name
UNION
SELECT g.name AS gallery_name, s.name AS exhibit_name
FROM galleries g
JOIN sculptures s ON s.gallery = g.name) t
GROUP BY t.gallery_name
ORDER BY exhibit_count DESC
GO
```

Резултат:

	gallery_name	exhibit_count
1	Asian	22
2	Classicism	14
3	Expressionism	13
4	Futurism	13
5	Abstract Expressionism	13
6	Rome	12
7	Gothic	11
8	Renaissance	7
9	Ancient Greece	7
10	Ancient Egypt	6

6.6. Да се изведе за всяка галерия авторът на най-много експонати в съответната галерия и техния брой. Резултатът да се сортира възходящо по името на галерията.

Заявка:

```

SELECT gallery, artist, COUNT(*) AS number_exhibits
FROM ( SELECT g.name AS gallery, p.name AS exhibit_name, artist
      FROM galleries g
      JOIN paintings p ON p.gallery = g.name
      UNION ALL
      SELECT g.name AS gallery, s.name AS exhibit_name, artist
      FROM galleries g
      JOIN sculptures s ON s.gallery = g.name ) d
GROUP BY gallery, artist
HAVING COUNT(*) >= ALL(SELECT COUNT(*)
                      FROM ( SELECT g.name AS gallery, p.name AS exhibit_name, artist
                            FROM galleries g
                            JOIN paintings p ON p.gallery = g.name
                            UNION ALL
                            SELECT g.name AS gallery, s.name AS exhibit_name, artist
                            FROM galleries g
                            JOIN sculptures s ON s.gallery = g.name ) d2
                      WHERE d.gallery = d2.gallery
                      GROUP BY gallery, artist)

ORDER BY gallery
GO

```

Резултат:

	gallery	artist	number_exhibits
1	Abstract Expressionism	Knud Agger	3
2	Asian	Kinji Akagawa	4
3	Classicism	Ivan Aivazovsky	3
4	Expressionism	Per Abramsen	3
5	Futurism	Abed Abdi	3
6	Gothic	Lambert-Sigisbert Adam	3
7	Renaissance	Tadeusz Ajdukiewicz	2
8	Rome	Yaacov Agam	2

6.7. За галериите с поне 2 зали, в които печалбата от посетителски карти е по-голяма от 1250 евро и броят на закупените карти е поне 23, да се изведе име на галерията, брой закупени карти, приходи, най-известен експонат и брой зали в галерията. Полученият резултат да се сортира низходящо по броя карти.

Заявка:

```
SELECT t.gallery, t.cards, t.profit, g.most_famous_exhibit, g.halls
FROM (SELECT gallery, SUM(price) AS profit, COUNT(*) AS cards
      FROM membership_cards
      GROUP BY gallery
      HAVING SUM(price) > 1250 AND COUNT(*) >= 23 ) t
JOIN galleries g ON t.gallery = g.name
WHERE g.halls > 2
ORDER BY cards DESC
GO
```

Резултат:

	gallery	cards	profit	most_famous_exhibit	halls
1	Ancient Greece	24	1304.05	Phaistos Disc	5
2	Renaissance	23	1362.68	Wild Geese	4
3	Rome	23	1358.74	Ateni Theotokos Church inscription	4
4	Ancient Egypt	23	1252.03	Seneb and his family	3

6.8. Да се намери художникът с най-голяма разлика в годините на създаване на най-ранната и на най-късната му картина, които притежава музея. Да се изведат освен името му, колко е разликата в годините и какъв е стилът, в който твори.

Заявка:

```
SELECT TOP 1 p1.artist, ((MAX(p1.year)) - (MIN(p1.year))) AS diff, (SELECT a.style FROM artists a
                                                                    WHERE a.name = p1.artist) AS style
FROM paintings p1
GROUP BY p1.artist
ORDER BY diff DESC, artist
GO
```

Резултат:

	artist	diff	style
1	Ivan Aivazovsky	38	Romantic

6.9. Медианата е число, разделящо горната половина на набор от данни от долната половина. Изведете стойността на този артефакт от Древна Гърция, чиято оценка се явява медиана като се посочат неговите номер, име, материал и оценка в евро.

Заявка:

```

SELECT a1.catalog_number, a1.name, a1.material, a1.[estimate(EURO)]
FROM artefacts a1
WHERE gallery LIKE 'Ancient Greece' AND
      (SELECT COUNT(*)
       FROM artefacts
       WHERE gallery = a1.gallery AND [estimate(EURO)] <= a1.[estimate(EURO)]) = (SELECT COUNT(*)
                                                                                     FROM artefacts
                                                                                     WHERE gallery = a1.gallery AND [estimate(EURO)] >= a1.[estimate(EURO)]);

GO

```

Резултат:

	catalog_number	name	material	estimate(EURO)
1	40013	Stele of Serapeitis	stone	324000

6.10. Да се изведе за всяка галерия броя на тези картини, на които площта е по-голяма или равна на площта на най-голямата картина на Josef Abel. Също така да се покаже за всяка галерия, каква е средната стойност на тези картини и резултата да се сортира възходящо по средната стойност на картините.

Заявка:

```

SELECT p1.gallery, COUNT(*) AS number_paintings, AVG(p1.[estimate(EURO)]) AS avg_price
FROM paintings p1
where (CAST(p1.[height(mm)] AS INT) * CAST(p1.[width(mm)] AS INT)) >=
      (SELECT MAX(CAST(p2.[height(mm)] AS INT) * CAST(p2.[width(mm)] AS INT))
       FROM paintings p2
       WHERE p2.artist LIKE 'Josef Abel' )
GROUP BY p1.gallery
HAVING COUNT(*) > 2
ORDER BY avg_price
GO

```

Резултат:

	gallery	number_paintings	avg_price
1	Expressionism	3	10133
2	Renaissance	5	14600
3	Asian	8	14787
4	Futurism	3	16000
5	Abstract Expressionism	3	23770

7. ОГРАНИЧЕНИЯ

Ограниченията са обекти в базите от данни, които налагат определени рестрикции на данните в дадена таблица. Могат да се задават при дефиниране на релационната схема или да се добавят след това.

7.1. Примери с деклариране на UNIQUE ключ в таблиците:


```
CREATE TABLE artists(
    name VARCHAR(50) NOT NULL,
    born SMALLINT,
    died SMALLINT,
    country CHAR(20) NOT NULL,
    style CHAR(30) NOT NULL,
    CONSTRAINT uk_artists UNIQUE (name)
);
```

```
CREATE TABLE galleries(
    name CHAR(30) NOT NULL,
    most_famous_exhibit VARCHAR(50) NOT NULL,
    [curator(EGN)] CHAR(10) NOT NULL,
    halls SMALLINT NOT NULL,
    CONSTRAINT uk_galleries UNIQUE (name)
);
```

```
CREATE TABLE curators(
    EGN CHAR(10) NOT NULL,
    name CHAR(30) NOT NULL,
    address VARCHAR(80) NOT NULL,
    date_of_birth DATE NOT NULL,
    university VARCHAR(50),
    date_of_appointment DATE NOT NULL,
    academic_title CHAR(30) NOT NULL,
    participation_in_exhibitions SMALLINT NOT NULL,
    CONSTRAINT UK_curators UNIQUE (EGN)
```

7.2. Примери с добавяне на ограничения за PRIMARY KEY

```
ALTER TABLE paintings ADD CONSTRAINT PK_paintings PRIMARY KEY(catalog_number);
ALTER TABLE artefacts ADD CONSTRAINT PK_artefacts PRIMARY KEY(catalog_number);
ALTER TABLE sculptures ADD CONSTRAINT PK_sculptures PRIMARY KEY(catalog_number);
ALTER TABLE artists ADD CONSTRAINT PK_artists PRIMARY KEY(name);
ALTER TABLE membership_cards ADD CONSTRAINT PK_membership_cards PRIMARY KEY(id);
ALTER TABLE galleries ADD CONSTRAINT PK_galleries PRIMARY KEY(name);
ALTER TABLE curators ADD CONSTRAINT PK_curators PRIMARY KEY(EGN);
```

7.3. Примери с добавяне на ограничения за FOREIGN KEY, при който се добавя и ключовата дума REFERENCES

```
ALTER TABLE paintings ADD CONSTRAINT FK_paintings_artists FOREIGN KEY(artist) REFERENCES artists(name);
ALTER TABLE paintings ADD CONSTRAINT FK_paintings_galleries FOREIGN KEY(gallery) REFERENCES galleries(name);
ALTER TABLE artefacts ADD CONSTRAINT FK_artefacts_galleries FOREIGN KEY(gallery) REFERENCES galleries(name);
ALTER TABLE sculptures ADD CONSTRAINT FK_sculptures_artists FOREIGN KEY(artist) REFERENCES artists(name);
ALTER TABLE sculptures ADD CONSTRAINT FK_sculptures_galleries FOREIGN KEY(gallery) REFERENCES galleries(name);
ALTER TABLE membership_cards ADD CONSTRAINT FK_membership_cards_galleries FOREIGN KEY(gallery) REFERENCES galleries(name);
ALTER TABLE galleries ADD CONSTRAINT FK_galleries_curator FOREIGN KEY([curator(EGN)]) REFERENCES curators(EGN);
```

7.4 Примери с добавяне на ограничения за CHECK ограничения

```

ALTER TABLE paintings ADD CONSTRAINT CHK_paintings_year CHECK (year >= 0);
ALTER TABLE paintings ADD CONSTRAINT CHK_paintings_height CHECK ([height(mm)] >= 0);
ALTER TABLE paintings ADD CONSTRAINT CHK_paintings_width CHECK ([width(mm)] >= 0);
ALTER TABLE paintings ADD CONSTRAINT CHK_paintings_gallery CHECK (gallery IN ('Ancient Greece', 'Ancient Egypt', 'Rome', 'Asian',
    'Gothic', 'Renaissance', 'Expressionism', 'Classicism', 'Futurism', 'Abstract Expressionism'));
ALTER TABLE paintings ADD CONSTRAINT CHK_paintings_estimate CHECK ([estimate(EURO)] >= 0);
ALTER TABLE artefacts ADD CONSTRAINT CHK_artefacts_portable CHECK (portable IN ('y', 'n'));
ALTER TABLE artefacts ADD CONSTRAINT CHK_artefacts_count CHECK (count >= 0);
ALTER TABLE artefacts ADD CONSTRAINT CHK_artefacts_gallery CHECK (gallery IN ('Ancient Greece', 'Ancient Egypt', 'Rome', 'Asian',
    'Gothic', 'Renaissance', 'Expressionism', 'Classicism', 'Futurism', 'Abstract Expressionism'));
ALTER TABLE artefacts ADD CONSTRAINT CHK_artefacts_estimate CHECK ([estimate(EURO)] >= 0);
ALTER TABLE artefacts ADD CONSTRAINT CHK_Person_Age CHECK ([estimate(EURO)] >= 0);
ALTER TABLE sculptures ADD CONSTRAINT CHK_sculptures_year CHECK (year >= 0);
ALTER TABLE sculptures ADD CONSTRAINT CHK_sculptures_area CHECK ([area_required(sq_cm)] >= 0);
ALTER TABLE sculptures ADD CONSTRAINT CHK_sculptures_gallery CHECK (gallery IN ('Ancient Greece', 'Ancient Egypt', 'Rome', 'Asian',
    'Gothic', 'Renaissance', 'Expressionism', 'Classicism', 'Futurism', 'Abstract Expressionism'));
ALTER TABLE sculptures ADD CONSTRAINT CHK_sculptures_estimate CHECK ([estimate(EURO)] >= 0);
ALTER TABLE artists ADD CONSTRAINT CHK_artists_born CHECK (born >= 0);
ALTER TABLE artists ADD CONSTRAINT CHK_artists_died CHECK (died >= 0);
ALTER TABLE artists ADD CONSTRAINT CHK_artists_lived CHECK (born < died);
ALTER TABLE membership_cards ADD CONSTRAINT CHK_membership_cards_gallery CHECK (gallery IN ('Ancient Greece', 'Ancient Egypt', 'Rome', 'Asian',
    'Gothic', 'Renaissance', 'Expressionism', 'Classicism', 'Futurism', 'Abstract Expressionism'));
ALTER TABLE membership_cards ADD CONSTRAINT CHK_membership_cards_price CHECK (price >= 0);
ALTER TABLE galleries ADD CONSTRAINT CHK_galleries_halls CHECK (halls >= 0);
ALTER TABLE curators ADD CONSTRAINT CHK_curators_exhibitions CHECK (participation_in_exhibitions >= 0);

```

8. ИЗГЛЕДИ И ИНДЕКСИ

Изгледи:

Те са виртуални таблици, които не съществуват физически върху диска и не съдържат собствени данни. Служат за ограничаване на достъпа до данните и улеснява тяхното обработване и извличане.

8.1. Изгледът `doctors_of_philosophy`, който извежда име и адрес на кураторите с титли Доктор по философия. Изгледът извежда и университета, които са завършили и също в колко изложби са взели участие.

```

CREATE VIEW doctors_of_philosophy AS
SELECT name, address, university, participation_in_exhibitions
FROM curators
WHERE curators.academic_title like 'Doctor of Philosophy'
GO

```

Заявки за тестване на изгледа:

Пример 1: Изведете броя на кураторите които са доктори по философия и са завършили Станфордския университет.

```

SELECT COUNT(name) as Graduated_Stanford
FROM doctors_of_philosophy
WHERE doctors_of_philosophy.university = 'Stanford University'
GO

```

Резултат:

	Graduated_Stanford
1	3

Пример 2: Изведете имената на кураторите взели участие в повече от 3 изложби и завършили Yale University.

```
(SELECT name as curator
FROM doctors_of_philosophy
WHERE doctors_of_philosophy.participation_in_exhibitions >3)
INTERSECT
(SELECT name as curator
FROM doctors_of_philosophy
WHERE doctors_of_philosophy.university LIKE 'Yale University')
GO
```

Резултат:

	curator
1	Carmen White

8.2. Изгледът Cities_England, който извежда име, адрес и град на посетителите от градове в Англия. Изгледът извежда и галерията, за която имат карта, чиято цена е по-голяма от 30 евро.

```
CREATE VIEW Cities_England AS
SELECT name, address, city, gallery, price
FROM membership_cards
WHERE city IN ('Bristol', 'London', 'Liverpool', 'Cambridge', 'Coventry', 'Glastonbury', 'Lancaster', 'Manchester', 'Oxford') and price > 30
GO
```

Заявки за тестване на изгледа:

Пример 1: Изведете всички хора, които живеят в Лондон и имат достъп до галерия Древен Египет.

```
SELECT name as visitor
FROM Cities_England
WHERE gallery = 'Ancient Egypt' and city = 'London'
GO
```

Резултат:

	visitor
1	Malachi Knox
2	Yvonne Person
3	Emelia Acosta

Пример 2: Изведете градовете, от които има повече от един човек с карта. Изведете и броя на клиентите.

```
SELECT city, COUNT(name) as Visitor_Number
FROM Cities_England
GROUP BY city
HAVING COUNT(name) > 1
GO
```

Резултат:

	city	Visitor_Number
1	Bristol	2
2	Liverpool	4
3	London	14
4	Manch...	2

8.3. Изгледът Sculptors, който извежда име на скулптурата, годината ѝ на създаване, име на скулптора, годината му на раждане и на колко е оценена творбата му.

```
CREATE VIEW Sculptors AS
SELECT sculptures.name AS sculpture, sculptures.year, sculptures.artist, artists.born AS Birthdate, sculptures.[estimate(EURO)] AS price
FROM artists JOIN sculptures ON sculptures.artist = artists.name
GO
```

Заявки за тестване на изгледа:

Пример 1: Изведете скулптурите, които са оценени на повече от 50 000.

```
SELECT sculpture
FROM Sculptors
WHERE price>50000
GO
```

Резултат:

	sculpture
1	Opettajani
2	Di Conway
3	Nierozpo...
4	Agora
5	Alchetron
6	Beacon
7	Monume...
8	Elek Lux

Пример 2: Изведете скулптора с най-много скулптури и техният брой.

```
SELECT artist, COUNT(sculpture) AS Number_of_sculptures
FROM Sculptors
GROUP BY artist
HAVING COUNT(sculpture) >= ALL(SELECT COUNT(sculpture) FROM sculptures GROUP BY artist)
GO
```

Резултат:

	artist	Number_of_sculptures
1	Per Abramsen	6

8.4. Изгледът Painters, който извежда име на картината, име на художника, стилът му и на колко е оценена творбата му.

```
CREATE VIEW Painters AS
SELECT paintings.name, paintings.artist, artists.style, paintings.[estimate(EURO)] AS price
FROM artists JOIN paintings ON artists.name = paintings.artist
GO
```

Заявки за тестване на изгледа:

Пример 1: Изведете името и цената на картините, чиито имена започват с 'T' и са само от три думи.

```
((SELECT DISTINCT name AS painting, price
FROM Painters
WHERE name LIKE 'T%' AND name LIKE '% % %'))
EXCEPT
(SELECT DISTINCT name AS painting, price
FROM Painters
WHERE name LIKE 'T%' AND name LIKE '% % % %'))
GO
```

Резултат:

	painting	price
1	The Last Emperor	14510
2	The Ninth Wave	380000
3	The Red Rook	54200
4	Tibetan girls eyes	13200

Пример 2: Изведете името и средната цена на картината с най-висока средна цена.

```
SELECT name as painting, AVG(price) as Average_Price
FROM Painters
GROUP BY name
HAVING AVG(price) >= ALL(SELECT AVG(price) FROM painters GROUP BY name)
GO
```

Резултат:

	painting	Average_Price
1	The Ninth Wave	380000

8.5. Изгледът Asian_gallery, който извежда имената на картините и скулптурите от галерията "Азия". Изгледът извежда и съответните скулптори и художници.

```
CREATE VIEW Asian_gallery
(sculpture, painting, sculptor, painter) AS
SELECT sculptures.name , paintings.name , sculptures.artist, paintings.artist
FROM sculptures JOIN paintings ON sculptures.gallery = paintings.gallery
WHERE sculptures.gallery = 'Asian'
WITH CHECK OPTION
GO
```

Заявки за тестване на изгледа:

Пример 1: Изведете всички художници от галерия "Азия" и броя на техните картини.

```
SELECT painter, COUNT(painting) as Number_of_paintings
FROM Asian_gallery
GROUP BY painter
GO
```

Резултат:

	painter	Number_of_paintings
1	Ai Xuan	12
2	Ai-Mitsu	12
3	An Gy...	12
4	Christ...	12
5	Jacqu...	6

Пример 2: Изведете скулпторите, чийто брой на скулптурите е по-голям от броя на картините на художникът с най-много творби.

```
SELECT sculptor
FROM Asian_gallery
GROUP BY sculptor
HAVING COUNT(sculpture) > ALL(SELECT COUNT(painting)
FROM asian_gallery
GROUP BY painter)
GO
```

Резултат:

	sculptor
1	Kinji Akagawa

Индекси:

Физически обект в базата от данни, който ускорява връщането на редове като използва указатели към адреси в паметта.

8.6. Създаваме индекс `index_paintings` на таблицата `paintings`.

```
CREATE INDEX index_paintings  
ON paintings(catalog_number, name, year, artist, technique, style, movement, [height(mm)], [width(mm)], gallery, [estimate(EURO)]);
```

8.7. Създаваме уникален индекс `index_painting` на таблицата

```
CREATE UNIQUE INDEX index_painting  
ON paintings(catalog_number);
```

8.8. Създаваме индекс `index_artefacts` на таблицата `artefacts`.

```
CREATE INDEX index_artefacts  
ON artefacts(catalog_number, name, century, discovery_site, civilization, material, portable, count, gallery, [estimate(EURO)]);
```

8.9. Създаваме уникален индекс `index_artefact` на таблицата `artefacts`.

```
CREATE UNIQUE INDEX index_artefact  
ON artefacts(catalog_number);
```

8.10. Създаваме индекс `index_sculptures` на таблицата `sculptures`.

```
CREATE INDEX index_sculptures  
ON sculptures(catalog_number, name, year, artist, material, country, [area_required(sq_cm)], gallery, [estimate(EURO)]);
```

8.11. Създаваме уникален индекс `index_sculpture` на таблицата `sculptures`.

```
CREATE UNIQUE INDEX index_sculpture  
ON sculptures(catalog_number);
```

8.12. Създаваме индекс `index_artists` на таблицата `artists`.

```
CREATE INDEX index_artists  
ON artists(name, born, died, country, style);
```

8.13. Създаваме индекс `index_membership_cards` на таблицата `membership_cards`.

```
CREATE INDEX index_membership_cards  
ON membership_cards(name, address, city, date_of_birth, date_of_expiry, gallery, price);
```

8.14. Създаваме индекс `index_galleries` на таблицата `galleries`.

```
CREATE INDEX index_galleries  
ON galleries(name, most_famous_exhibit, [curator(EGN)], halls);
```

8.15. Създаваме индекс `index_curators` на таблицата `curators`.

```
CREATE INDEX index_curators  
ON curators(EGN, name, address, date_of_birth, university, date_of_appointment, academic_title, participation_in_exhibitions);
```


9. ТРИГЕРИ

9.1. Следващите три тригера запълват лог таблица, като при всяка промяна на оценъчна цена на картина, скулптура или артефакт се запълва кортеж с уникално id на log-a, каталожен номер на артефакта, новата и старата оценъчна цена, както и дата на съответната промяна.

```
CREATE OR ALTER TRIGGER tr_estimate_info_paintings ON paintings FOR UPDATE
AS
DECLARE @new INT = (SELECT TOP(1) [estimate(EURO)] FROM inserted)
DECLARE @old INT = (SELECT TOP(1) [estimate(EURO)] FROM deleted)

DECLARE @catalog_number INT = (SELECT catalog_number from inserted)

INSERT INTO logs (catalog_number, new_estimate, old_estimate, date_of_change) VALUES
(@catalog_number, @new, @old, GETDATE())
DECLARE @affected_rows INT = (SELECT COUNT(log_id) FROM logs WHERE catalog_number is NULL)
IF (@affected_rows > 0)
BEGIN
    ROLLBACK
    RAISERROR('Update on invalid articul.', 16, 1)
    RETURN
END
GO

CREATE OR ALTER TRIGGER tr_estimate_info_sculptures ON sculptures FOR UPDATE
AS
DECLARE @new INT = (SELECT TOP(1) [estimate(EURO)] FROM inserted)
DECLARE @old INT = (SELECT TOP(1) [estimate(EURO)] FROM deleted)

DECLARE @catalog_number INT = (SELECT catalog_number from inserted)

INSERT INTO logs (catalog_number, new_estimate, old_estimate, date_of_change) VALUES
(@catalog_number, @new, @old, GETDATE())
DECLARE @affected_rows INT = (SELECT COUNT(log_id) FROM logs WHERE catalog_number is NULL)
IF (@affected_rows > 0)
BEGIN
    RAISERROR('Update on invalid articul.', 16, 1)
    ROLLBACK
    RETURN
END
GO

CREATE OR ALTER TRIGGER tr_estimate_info_artefacts ON artefacts FOR UPDATE
AS
DECLARE @new INT = (SELECT TOP(1) [estimate(EURO)] FROM inserted)
DECLARE @old INT = (SELECT TOP(1) [estimate(EURO)] FROM deleted)

DECLARE @catalog_number INT = (SELECT catalog_number from inserted)

INSERT INTO logs (catalog_number, new_estimate, old_estimate, date_of_change) VALUES
(@catalog_number, @new, @old, GETDATE())
DECLARE @affected_rows INT = (SELECT COUNT(log_id) FROM logs WHERE catalog_number is NULL)
IF (@affected_rows > 0)
BEGIN
    RAISERROR('Update on invalid exhibit.', 16, 1)
    ROLLBACK
    RETURN
END
GO
```

9.2. Следващия тригер запълва email таблица, като при всяка изтъркана посетителска карта, ще служи за изпращане на email на собственика със заглавна тема, която казва, че картата му е изтекла и съдържание - датата на изтичане на картата и галерията за която е била издадена. Тригера използва процедура, която изважда всички изтекли посетителски карти от съответната таблица и ги изтрива една по една, за да може тригера да обработи една по една заявките за изтриване, като реално той не ги изтрива, а ги прави неактивни и едновременно добавя информация в лог файла.


```

CREATE OR ALTER TRIGGER tr_log_emails ON membership_cards INSTEAD OF DELETE
AS
DECLARE @del_id INT = (SELECT TOP(1) id FROM deleted)
DECLARE @recipient VARCHAR(80) = (SELECT TOP(1) name FROM deleted)
DECLARE @gallery VARCHAR(30) = (SELECT TOP(1) gallery FROM deleted)
DECLARE @bit BIT = (SELECT TOP(1) is_active FROM deleted)
DECLARE @card_number INT = (SELECT TOP(1) id FROM deleted)
IF(@bit = 1)
    INSERT INTO notification_emails(card_number, recipient, subject, body) VALUES
    (
        @card_number,
        @recipient,
        'Expired membership card for gallery: ' + @gallery,
        'On ' + CONVERT(varchar(30), CONVERT(date, GETDATE(), 20)) + ' your membership card has expired. Thanks for visiting us!'
    )
    UPDATE membership_cards SET is_active = 0 where id = @del_id
GO

CREATE OR ALTER PROCEDURE pr_delete_expired_cards
AS
BEGIN
DECLARE @local_table_variable TABLE
(
    id INT
)
INSERT INTO @local_table_variable
SELECT id FROM membership_cards
WHERE date_of_expiry < GETDATE()

DECLARE @i INT
SELECT @i = min(id) from @local_table_variable
DECLARE @max INT
SELECT @max = max(id) from @local_table_variable

WHILE @i <= @max BEGIN
    DELETE FROM membership_cards WHERE id = (SELECT id FROM @local_table_variable WHERE id = @i)
    SET @i = @i + 1
END
END
GO

```

Тестване на първите три тригера:

```

UPDATE paintings SET [estimate(EURO)] += 5000 WHERE catalog_number = 10000
UPDATE paintings SET [estimate(EURO)] += 15000 WHERE catalog_number = 10000
SELECT * FROM logs

```

Резултат:

	log_id	catalog_number	new_estimate	old_estimate	date_of_change
1	1	10000	16000	11000	2020-05-04
2	2	10000	31000	16000	2020-05-04

В тригера има имплементиран допълнителен алгоритъм за справяне с грешките:

```

UPDATE artefacts SET [estimate(EURO)] -= 9200 WHERE catalog_number = 10001

```

120 %

Messages

```

(1 row affected)
Msg 50000, Level 16, State 1, Procedure tr_estimate_info_artefacts, Line 14 [Batch Start Line 870]
Update on invalid exhibit.
Msg 3609, Level 16, State 1, Line 871
The transaction ended in the trigger. The batch has been aborted.

```

Тестване на email_log тригера:

```

exec pr_delete_expired_cards
select * from notification_emails

```

	id	recipient	subject	body
1	1	Alanna West	Expired membership card for gallery: Abstract Expressionism	On 2020-05-02 your membership card has expired. Thanks for visiting us!
2	2	Stefan Mills	Expired membership card for gallery: Asian	On 2020-05-02 your membership card has expired. Thanks for visiting us!
3	3	Fay Keith	Expired membership card for gallery: Abstract Expressionism	On 2020-05-02 your membership card has expired. Thanks for visiting us!
4	4	Morgan Rodrigues	Expired membership card for gallery: Expressionism	On 2020-05-02 your membership card has expired. Thanks for visiting us!
5	5	Kade Summers	Expired membership card for gallery: Rome	On 2020-05-02 your membership card has expired. Thanks for visiting us!
6	6	Habibah Cantu	Expired membership card for gallery: Rome	On 2020-05-02 your membership card has expired. Thanks for visiting us!
7	7	Maia Hamington	Expired membership card for gallery: Expressionism	On 2020-05-02 your membership card has expired. Thanks for visiting us!

	id	name	address	city	date_of_birth	date_of_expiry	gallery	price	is_active
114	30113	Rahim James	4 Blackwall Way	London	1975-08-16	2025-02-09	Rome	28.64	1
115	30114	Eddison Brady	43 Maple Drive	Walsall	1999-06-20	2020-06-01	Asian	52.44	1
116	30115	Amritpal Casey	Old Town House	Bishops Itching...	1963-07-08	2020-05-03	Gothic	20.10	0
117	30116	Aamir Finnegan	22 Admiralty Road	Southbourne	1975-06-25	2021-12-01	Renaissance	47.80	1
118	30117	Lily Reese	16 Opal Way	Bishops Cleeve	1963-05-06	2020-02-04	Expressionism	56.51	0

11. БЪДЕЩИ ПЕРСПЕКТИВИ НА ПРОДУКТА

11.1. Информацията за имейлите на клиентите на музея ще бъде интегрирана към базата данни. Това ще доведе до забързване и улесняване на комуникацията с тях.

11.2. Добавяне на процедури за автоматизирано изпращане на email-и за всяко създаване на нова галерия, обновяване или пренареждане на налична такава.

Имейлите могат да съдържат името

на куратора, организиран подреждането на галерията, името на новата галерия или друга маркетингова информация, която може да привлече вниманието на настоящи и бивши клиенти на музея.

11.3. Добавяне на информация за персонала на музея. Ще бъде включена информация за техния график, дата на назначаване, заплата и други. Ще се поддържа информация за периодите, в които всеки от персонала е използвал отпуска, както и за неизползваните дни отпуска на всеки един от служителите.

11.4. За защита от грешки или злонамерена употреба в базата данни може да се предприемат допълнителни защити като се добавят тригери, които се грижат за soft delete на кортежи в таблиците за експонати или лог таблиците. Също така може да се добави допълнителна валидация на имената, които се въвеждат, например да не съдържат цифри или забранени символи, за осигуряване на валидност и коректност на данните.

