

Museum Database Manager

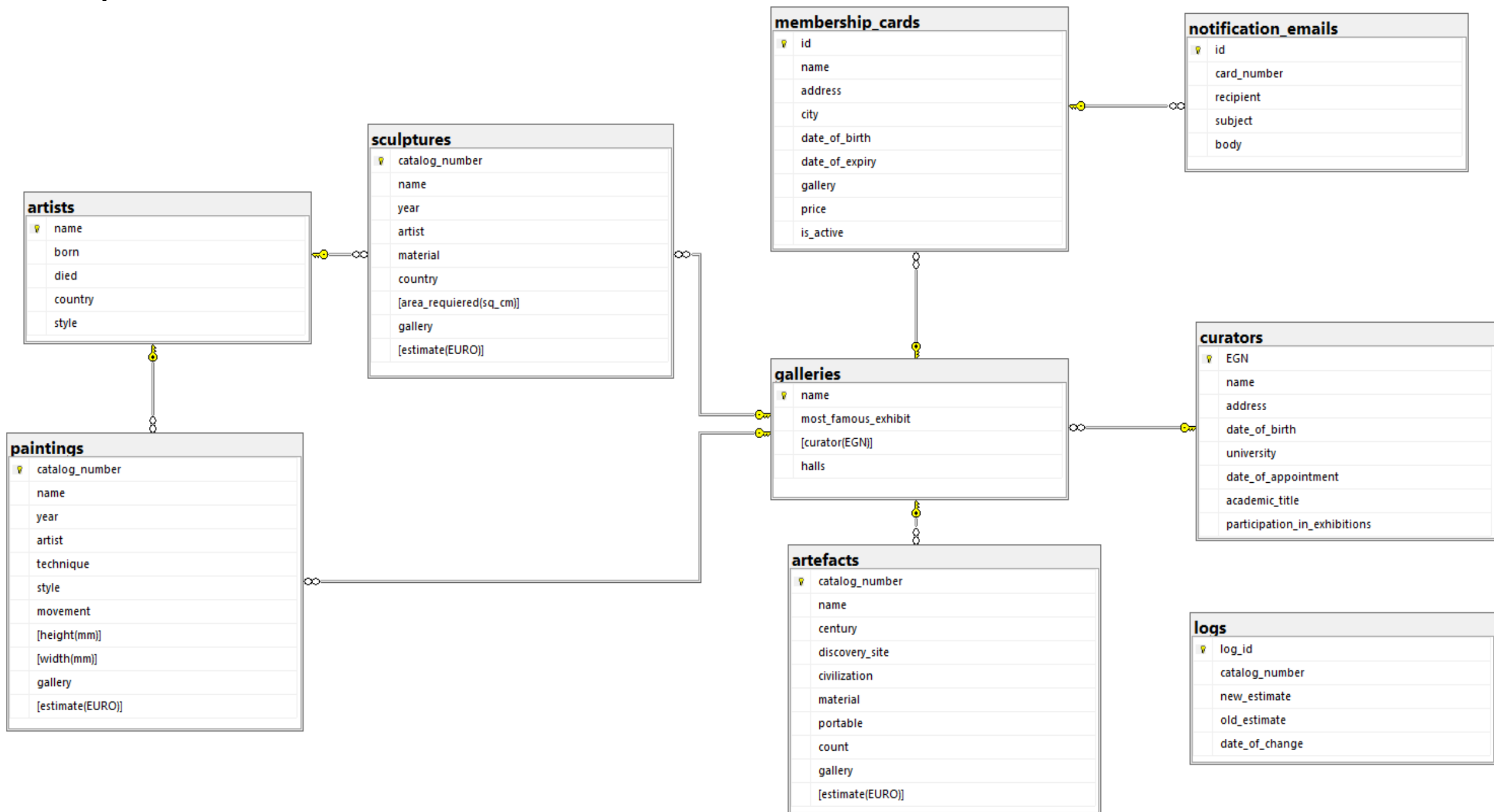


Изготвено от:
Таня Желева
Десислава Желязкова
Калоян Николов
Светлин Попиванов
Андрей Стое

Потенциални проблеми, които разработената база от данни може да решава:

- Проблеми свързани с организацията на музея и предлаганите услуги;
- Реклама и информиране на потенциални клиенти за експонати и изложби;
- В случай на кражба и или увреждане на експонати;
- Лесно конфигуриране на гостуващи изложби – предназначени за изложения в чужди музеи и държави;
- Помощ на изследователи и научни звена в изследвания в областта на изкуството и археологията;
- Лесно отчитане на посетителския интерес и приходите на музея;
- Източник на нови знания и база за изследвания на студенти и ученици, по изкуство, история и археология.

Диаграма на базата от данни



Прости Заявки

Пример 1:

Изведете имената и адресите на всички хора с абонаментни карти, чиито адреси започват с 2.

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

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

Пример 2:

Изведете 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'

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

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

Заявки върху две или повече релации

Пример 1:

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

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

Резултат:

	gallery
1	Ancient Egypt
2	Ancient Greece
3	Asian
4	Rome

Пример 2:

Изведете 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'.

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

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

Пример 3:

Изведете имената на всички художници, които са от Италия и стилът на картините им е 'Landscape'.

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

Резултат:

	artist
1	Giuseppe Abbati

Подзаявки

Пример 1:

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

```
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

Пример 2:

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

```
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

Съединения

Пример 1:

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

```
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

Пример 2

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

```
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

Групиране и агрегация

Пример 1:

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

```
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

Пример 2:

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.

```
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 Expr...	3	23770

Пример 3:

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.

```
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

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.

Пример с деклариране на UNIQUE ключ в таблиците.

```
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 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)  
);
```

- Примери с добавяне на ограничения за PRIMARY KEY, за FOREIGN KEY, при който се добавя и ключовата дума REFERENCES и също така за CHECK ограничения.

```
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);

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);

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',
```

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.
- Пример 1 за изглед, който извежда информация за посетители от градове в Англия.

```
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
```

Пример 1: 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

- Пример 2 с изглед, който извежда информация за картините и художниците им.

```
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
```

Пример 2: Изведете името и цената на картините, чиито имена започват с 'T' 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 3 с изглед, който извежда информация за картините и скулптурите, и техните творци, в тематичната галерия “Азия”.

```
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
```

Пример 3: 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.
- Примери за индекси:

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

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

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

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

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

- 
- Примери с UNIQUE индекси за някои атрибути в таблиците.

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

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


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

Тригери

Тригерът запълва email таблица, като при всяка изтъркана посетителска карта, ще изпраща email на собственика с подходящо съобщение включващо дата на изтичане и галерия, за която е била издадена.

```
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)
IF(@bit = 1)
    INSERT INTO notification_emails(recipient, subject, body) VALUES
    (
        @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



За да може да прихванем всички посетителски карти, които са изтекли сме написали процедура, която да ги събира в таблица и да ги обхожда ред по ред след което да ги изтрива една по една. Реално тригера ще се грижи за това те да не се изтриват, а само да се отбелязват като неактивни и да се изпраща email.

```
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
```

РЕЗУЛТАТ:

след извикване на процедурата всички изтекли посетителски карти ще бъдат отбелязани като неактивни и за всяка неактивна карта ще бъде попълнен кортеж в лог таблицата

```
exec pr_delete_expired_cards  
select * from notification_emails
```

121 %

Results Messages

	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 Harrington	Expired membership card for gallery: Expressionism	On 2020-05-02 your membership card has expired. Thanks for visiting us!

За неактивните карти ще се сетва бит, който ще показва че са изтекли


	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

Тригерите 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 цена.

```
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
```



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

	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.

Бъдещи перспективи на продукта

- Добавяне на детайлна информация за контакти с клиентите – e-mail, GSM и др.
- Автоматизирано информироване за нови събития и експонати.
- Добавяне на информация за персонала на музея.
- За защита от грешки или злонамерена употреба в базата от данни.

Благодарим за вниманието!

