

Petshop

1. Резултати от създадените функции

- Проверка дали питомец е достъпен и се намира в приюта

```
CREATE FUNCTION fn_IsPetAvailable (@PetID INT)
RETURNS BIT
AS
BEGIN
    DECLARE @IsAdopted BIT;

    SELECT @IsAdopted = IsAdopted
    FROM Pets
    WHERE PetID = @PetID;

    RETURN CASE WHEN @IsAdopted = 0 THEN 1 ELSE 0 END;
END;

SELECT dbo.fn_IsPetAvailable(3) AS IsAvailable;
SELECT
    p.PetID,
    p.Name,
    p.IsAdopted,
    dbo.fn_IsPetAvailable(p.PetID) AS IsAvailable
FROM Pets p;

SELECT *
FROM Pets
WHERE dbo.fn_IsPetAvailable(PetID) = 1;
```

	PetID	Name	Age	Breed	Gender	Description	IsAdopted	CategoryID	ShelterID
1	1	Buddy	3	Golden Retriever	Male	Friendly and playful.	0	1	1
2	2	Rex	5	German Shepherd	Male	Loyal and trained.	0	1	2
3	3	Lucky	2	Beagle	Male	Energetic and curious.	0	1	3
4	4	Bella	4	Husky	Female	Beautiful blue eyes.	0	1	4
5	6	Luna	2	Siamese	Female	Calm and affectionate.	0	2	1
6	7	Shadow	3	Black Cat	Male	Quiet and gentle.	0	2	2
7	8	Mimi	1	Shorthair	Female	Very playful.	0	2	3
8	10	Kitty	4	Domestic	Female	Loves attention.	0	2	5
9	12	Brownie	2	Brown Rabbit	Male	Calm.	0	3	2
10	13	Sky	1	Parrot	Male	Talkative.	0	4	3

Фигура 1. Резултат на първи 10 запроса от функция fn_IsPetAvailable

- Кой приют има наймного дарения

```
CREATE FUNCTION fn_TotalDonations (@ShelterID INT)
RETURNS DECIMAL(10,2)
AS
BEGIN
    DECLARE @Total DECIMAL(10,2);

    SELECT @Total = SUM(Amount)
    FROM Donations
    WHERE ShelterID = @ShelterID;

    RETURN ISNULL(@Total, 0);
END;

SELECT dbo.fn_TotalDonations(2) AS TotalDonations;
```

	TotalDonations
1	980.50

Фигура 2. Резултат запроса от функция fn_TotalDonations на втори Shelter_ID

```
SELECT
    s.ShelterID,
    s.ShelterName,
    dbo.fn_TotalDonations(s.ShelterID) AS TotalDonations
FROM Shelters s
ORDER BY TotalDonations DESC;
```

	ShelterID	ShelterName	TotalDonations
1	4	AnimalHope	1638.00
2	5	Life for Paws	1552.50
3	1	HappyPaws Shelter	1300.00
4	3	Green Valley	1012.00
5	2	Second Chance	980.50

Фигура 3. Резултат запроса от функция fn_TotalDonations по вид на топ

```
SELECT
    ShelterID,
    ShelterName
FROM Shelters
WHERE dbo.fn_TotalDonations(ShelterID) > 200;
```

	ShelterID	ShelterName
1	1	HappyPaws Shelter
2	2	Second Chance
3	3	Green Valley
4	4	AnimalHope
5	5	Life for Paws

Фигура 4. Резултат запроса от функция fn_TotalDonations на вид кое има повече от някой сума

- **Функция която изчислява колко животни има в приюта**

```
CREATE FUNCTION fn_PetCount (@ShelterID INT)
RETURNS INT
AS
BEGIN
    DECLARE @Count INT;

    SELECT @Count = COUNT(*)
    FROM Pets
    WHERE ShelterID = @ShelterID;

    RETURN @Count;
END;
```

```
SELECT dbo.fn_PetCount(1) AS PetCount;
```

	PetCount
1	8

Фигура 5. Резултат запроса от функция *fn_PetCount* на вид колко има определен приют

```
SELECT
    s.ShelterID,
    s.ShelterName,
    dbo.fn_PetCount(s.ShelterID) AS PetCount
FROM Shelters s
ORDER BY PetCount DESC;
```

	ShelterID	ShelterName	PetCount
1	1	HappyPaws Shelter	8
2	2	Second Chance	8
3	3	Green Valley	6
4	4	AnimalHope	6
5	5	Life for Paws	5

Фигура 6. Резултат запроса от функция *fn_PetCount* на вид на топ

```
SELECT
    ShelterID,
    ShelterName
FROM Shelters
WHERE dbo.fn_PetCount(ShelterID) >= 5;
```

	ShelterID	ShelterName
1	1	HappyPaws Shelter
2	2	Second Chance
3	3	Green Valley
4	4	AnimalHope
5	5	Life for Paws

Фигура 7. Резултат запроса от функция *fn_PetCount* който има повече от 5 питомца

2. Резултати от създадените процедури

- Процедура за създаване заявка в режим на очакване

```
CREATE PROCEDURE sp_CreateAdoptionRequest
    @UserID INT,
    @PetID INT
AS
BEGIN
    SET NOCOUNT ON;

    IF (dbo.fn_IsPetAvailable(@PetID) = 0)
    BEGIN
        RAISERROR('This pet is already adopted.', 16, 1);
        RETURN;
    END

    INSERT INTO AdoptionRequests (UserID, PetID, Status)
    VALUES (@UserID, @PetID, 'Pending');
END;

exec sp_CreateAdoptionRequest 3, 12;

SELECT * FROM AdoptionRequests ORDER BY RequestID DESC;
```

26	30	3	12	2025-11-16 18:45:02.323	Pending	NULL
----	----	---	----	-------------------------	---------	------

Фигура 8. Резултат запроса от процедура *sp_CreateAdoptionRequest*

- Процедура за апрув на заяката

```
CREATE PROCEDURE sp_ApproveAdoptionRequest
    @RequestID INT
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @PetID INT;

    SELECT @PetID = PetID
    FROM AdoptionRequests
    WHERE RequestID = @RequestID;

    UPDATE AdoptionRequests
    SET Status = 'Approved'
    WHERE RequestID = @RequestID;

    UPDATE Pets
    SET IsAdopted = 1
    WHERE PetID = @PetID;
END;

exec sp_ApproveAdoptionRequest 5;

select * from AdoptionRequests where RequestID = 5;
```

	RequestID	UserID	PetID	RequestDate	Status	ApprovedDate
1	5	1	5	2025-11-15 22:37:38.967	Approved	NULL

Фигура 9. Резултат запроса от процедура *sp_ApproveAdoptionRequest* на апрув на заявката

- Процедура на отклоняване на заявката

```
CREATE PROCEDURE sp_RejectAdoptionRequest
    @RequestID INT
AS
BEGIN
    SET NOCOUNT ON;

    UPDATE AdoptionRequests
    SET Status = 'Rejected'
    WHERE RequestID = @RequestID;
END;
```

```
exec sp_RejectAdoptionRequest 7;
```

```
SELECT * FROM AdoptionRequests WHERE RequestID = 7;
```

	RequestID	UserID	PetID	RequestDate	Status	ApprovedDate
1	7	3	7	2025-11-15 22:37:38.967	Rejected	NULL

Фигура 10. Резултат запроса от процедура *sp_RejectAdoptionRequest* на апрув на заявката

3. Резултати от създадените тригери

- Съхранява и показва дарения които биха направяни и от коя дата

```
CREATE TABLE DonationLog (
    LogID INT IDENTITY PRIMARY KEY,
    DonationID INT NOT NULL,
    LogMessage NVARCHAR(255),
    LogDate DATETIME DEFAULT GETDATE()
);
```

```
CREATE TRIGGER trg_LogDonation
ON Donations
AFTER INSERT
AS
BEGIN
    INSERT INTO DonationLog (DonationID, LogMessage)
    SELECT DonationID, 'New donation received: ' + CAST(Amount AS NVARCHAR)
    FROM inserted;
END;
```

```
SELECT * FROM DonationLog ORDER BY LogID DESC;
```

	LogID	DonationID	LogMessage	LogDate
1	91	38	New donation received: 25.00	2025-11-18 13:03:22.807
2	90	39	New donation received: 40.50	2025-11-18 13:03:22.807
3	89	40	New donation received: 15.00	2025-11-18 13:03:22.807
4	88	41	New donation received: 100.00	2025-11-18 13:03:22.807
5	87	42	New donation received: 60.00	2025-11-18 13:03:22.807
6	86	43	New donation received: 75.50	2025-11-18 13:03:22.807
7	85	44	New donation received: 20.00	2025-11-18 13:03:22.807
8	84	45	New donation received: 35.00	2025-11-18 13:03:22.807
9	83	46	New donation received: 50.00	2025-11-18 13:03:22.807
10	82	47	New donation received: 90.00	2025-11-18 13:03:22.807
11	81	48	New donation received: 120.00	2025-11-18 13:03:22.807
12	80	49	New donation received: 30.00	2025-11-18 13:03:22.807
13	79	50	New donation received: 45.00	2025-11-18 13:03:22.807

Фигура 11. Резултат запроса от триггер *trg_LogDonation*

- Триггер за проверка на възраст от питомца, не може да бъде по малко от нула

```
CREATE TRIGGER trg_CheckPetAge
ON Pets
FOR INSERT, UPDATE
AS
BEGIN
    IF EXISTS (SELECT 1 FROM inserted WHERE Age < 0)
    BEGIN
        RAISERROR('Age cannot be negative.', 16, 1);
        ROLLBACK TRANSACTION;
        RETURN;
    END
END;
```

```
INSERT INTO Pets (Name, Age, Breed, Gender, Description, CategoryID, ShelterID)
VALUES ('BadPet', -2, 'Unknown', 'Female', 'Invalid age', 1, 1);
```

```
сообщение: 50000, уровень: 16, состояние: 1, процедура: trg_Che
Age cannot be negative.
Сообщение 3609, уровень 16, состояние 1, строка 40
Транзакция завершилась в триггере. Выполнение пакета прервано.
```

Фигура 12. Резултат от опитване да инсертнем дата

```
INSERT INTO Pets (Name, Age, Breed, Gender, Description, CategoryID, ShelterID)
VALUES ('TestPet', 3, 'TestBreed', 'Male', 'Healthy pet', 1, 1);
```

```
SELECT * FROM Pets WHERE Name = 'TestPet';
```

	PetID	Name	Age	Breed	Gender	Description	IsAdopted	CategoryID	ShelterID
1	34	TestPet	3	TestBreed	Male	Healthy pet	0	1	1

Фигура 12. Резултат от инсерт на дата(успешно, *trg_CheckPetAge*)