

## Project – SQL Statements

R Marx

Lecturer: Linda Redelinghuys

36581852

GitHub: <https://github.com/Tanaille/cmpg223-project>

## Table of Contents

1. Table Creation .....	3
1.1 City .....	3
1.2 User Type .....	3
1.3 Animal Type .....	3
1.4 Users .....	3
1.5 Passwords .....	3
1.6 Adopter .....	4
1.7 Animal .....	4
1.8 Adoption .....	4
2. SELECT Queries.....	5
2.1 User authentication .....	5
2.2 Get NULL values .....	5
2.3 Check if record exists .....	5
2.4 Value retrieval.....	5
2.5 Populate and filter form controls .....	7
2.6 Reporting.....	8
3. Maintenance Queries (INSERT, UPDATE, DELETE) .....	10
3.1 Adopters.....	10
3.2 Adoptions.....	10
3.3 Animals.....	10
3.4 Animal Types.....	11
3.5 Cities.....	11
3.6 Passwords .....	11
3.7 Users .....	11
3.8 User Types.....	12

## 1. Table Creation

### 1.1 City

```
CREATE TABLE CITY (  
  City_ID int IDENTITY(1,1) NOT NULL PRIMARY KEY,  
  City_Name varchar(35)  
);
```

### 1.2 User Type

```
CREATE TABLE USER_TYPE (  
  User_Type tinyint IDENTITY(1,1) NOT NULL PRIMARY KEY,  
  Type_Description varchar(20)  
);
```

### 1.3 Animal Type

```
CREATE TABLE ANIMAL_TYPE (  
  Type_Number int IDENTITY(1,1) NOT NULL PRIMARY KEY,  
  Type_Name varchar(35)  
);
```

### 1.4 Users

```
CREATE TABLE USERS (  
  User_Number int IDENTITY(1,1) NOT NULL,  
  User_Type tinyint,  
  User_Name varchar(35),  
  User_Surname varchar(35),  
  CONSTRAINT PK_USERS PRIMARY KEY (User_Number),  
  CONSTRAINT FK_USERS_USER_TYPE FOREIGN KEY (User_Type) REFERENCES USER_TYPE (User_Type)  
  ON DELETE SET NULL  
);
```

### 1.5 Passwords

```
CREATE TABLE PASSWORDS (  
  User_Number int NOT NULL,  
  Hash varchar(100),  
  Salt varchar(100),  
  CONSTRAINT PK_PASSWORDS PRIMARY KEY (User_Number),  
  CONSTRAINT FK_PASSWORDS_USER FOREIGN KEY (User_Number) REFERENCES USERS  
  (User_Number) ON DELETE CASCADE  
);
```

## 1.6 Adopter

```
CREATE TABLE ADOPTER (  
  Adopter_Number          int IDENTITY(1,1) NOT NULL,  
  Adopter_Surname         varchar(35),  
  Adopter_Name            varchar(35),  
  Adopter_Contact_Number  char(10),  
  Adopter_Street_Number   varchar(6),  
  Adopter_Street_Name     varchar(50),  
  City_ID                 int,  
  CONSTRAINT PK_ADOPTER PRIMARY KEY (Adopter_Number),  
  CONSTRAINT FK_ADOPTER_CITY FOREIGN KEY (City_ID) REFERENCES CITY (City_ID) ON DELETE SET  
  NULL  
);
```

## 1.7 Animal

```
CREATE TABLE ANIMAL (  
  Animal_Number          int IDENTITY(1,1) NOT NULL,  
  Animal_Name            varchar(35),  
  Type_Number            int,  
  Add_Date               date,  
  CONSTRAINT PK_ANIMAL PRIMARY KEY (Animal_Number),  
  CONSTRAINT FK_ANIMAL_ANIMAL_TYPE FOREIGN KEY (Type_Number) REFERENCES ANIMAL_TYPE  
  (Type_Number) ON DELETE SET NULL  
);
```

## 1.8 Adoption

```
CREATE TABLE ADOPTION (  
  Adoption_Number        int IDENTITY(1,1) NOT NULL,  
  Adopter_Number         int,  
  Animal_Number          int,  
  User_Number            int,  
  Result_Description     varchar(10),  
  Adoption_Date          date,  
  CONSTRAINT PK_ADOPTION PRIMARY KEY (Adoption_Number),  
  CONSTRAINT FK_ADOPTION_ADOPTER FOREIGN KEY (Adopter_Number) REFERENCES ADOPTER  
  (Adopter_Number) ON DELETE CASCADE,  
  CONSTRAINT FK_ADOPTION_ANIMAL FOREIGN KEY (Animal_Number) REFERENCES ANIMAL  
  (Animal_Number) ON DELETE CASCADE,  
  CONSTRAINT FK_ADOPTION_USERS FOREIGN KEY (User_Number) REFERENCES USERS  
  (User_Number)  
);
```

## 2. SELECT Queries

### 2.1 User authentication

```
SELECT USERS.User_Name, PASSWORDS.Hash, PASSWORDS.Salt
FROM USERS
INNER JOIN PASSWORDS
ON USERS.User_Number = PASSWORDS.User_Number
WHERE USERS.User_Name = @userName;
```

### 2.2 Get NULL values

```
SELECT *
FROM table
WHERE + column IS NULL;
```

### 2.3 Check if record exists

```
SELECT TOP 1 column
FROM table
WHERE column = @value;
```

### 2.4 Value retrieval

#### **Animal type number from animal type name**

```
SELECT Type_Number
FROM ANIMAL_TYPE
WHERE Type_Name = @typeName;
```

#### **Animal type number from animal name**

```
SELECT Type_Number
FROM ANIMAL
WHERE Animal_Name = @name AND Type_Number IS NOT NULL;
```

#### **Animal number from animal name**

```
SELECT Animal_Number
FROM ANIMAL
WHERE Animal_Name = @animalName;
```

#### **Animal type name from animal type number**

```
SELECT Type_Name
FROM ANIMAL_TYPE
WHERE Type_Number = @typeName;
```

#### **User type number from user type name**

```
SELECT User_Type
FROM USER_TYPE
WHERE Type_Description = @typeName;
```

#### **User number from user name**

```
SELECT User_Number
FROM USERS
WHERE User_Name = @userName AND User_Surname = @userSurname;
```

#### **User surname from user name**

```
SELECT User_Surname
FROM USERS
WHERE User_Name = @userName;
```

#### **City ID from city name**

```
SELECT City_ID
FROM CITY
WHERE City_Name = @cityName;
```

#### **Adopter surname from adopter name**

```
SELECT Adopter_Surname
FROM ADOPTER
WHERE Adopter_Name = @adopterName;
```

#### **Adopter number from adopter name**

```
SELECT Adopter_Number
FROM ADOPTER
WHERE Adopter_Name = @adopterName AND Adopter_Surname = @adopterSurname;
```

#### **Adoption number from animal number**

```
SELECT Adoption_Number
FROM ADOPTION
WHERE Animal_Number = @animalNumber;
```

#### **Adoption number from adopter name**

```
SELECT Adoption_Number
FROM ADOPTION
WHERE Adopter_Number = @adopterNumber;
```

#### **User name and surname from user number**

```
SELECT User_Name, User_Surname  
FROM USERS  
WHERE User_Number = @userNumber;
```

#### **User type description from user name**

```
SELECT Type_Description  
FROM USER_TYPE  
WHERE USERS.User_Name = @userName;
```

#### **User type number from user name**

```
SELECT User_Type  
FROM USERS  
WHERE User_Name = @userName;
```

#### **User type description from type number**

```
SELECT Type_Description  
FROM USER_TYPE  
WHERE User_Type = @typeNumber;
```

#### **Authenticate a user**

```
SELECT USERS.User_Name, PASSWORDS.Hash, PASSWORDS.Salt  
FROM USERS  
INNER JOIN PASSWORDS  
ON USERS.User_Number = PASSWORDS.User_Number  
WHERE USERS.User_Name = @userName;
```

## **2.5    [Populate and filter form controls](#)**

#### **Populate combobox**

```
SELECT column  
FROM table  
ORDER BY column ASC;
```

#### **Populate datagridview**

```
SELECT columns  
FROM tables[0];
```

```
SELECT columns  
FROM tables[0] a, tables[1] b WHERE a.key = b.key;
```

```
SELECT columns
```

```
FROM tables[0] a, tables[1] b, tables[2] c
WHERE a.key = b.key AND b.key = c.key;
```

```
SELECT columns
FROM ADOPTION a, ADOPTER b, USERS c, ANIMAL d
WHERE a.Adopter_Number = b.Adopter_Number AND a.User_Number = c.User_Number AND
a.Animal_Number = d.Animal_Number;
```

### **Populate listbox**

```
SELECT columns[0]
FROM table a
ORDER BY columns[0] ASC;
```

```
SELECT columns[0], columns[1]
FROM table[0] a, table[1] b
WHERE a.key = b.key
ORDER BY columns[0] ASC;
```

### **Filter listbox**

```
SELECT column
FROM table
WHERE column
LIKE '%' + filter + '%'
ORDER BY column ASC;
```

## **2.6 Reporting**

### **Animals adopter per time period**

```
SELECT a.Animal_Name, a.Animal_Number, b.Adopter_Name, c.Adoption_Date, d.User_Name
FROM ANIMAL a, ADOPTER b, ADOPTION c, USERS d
WHERE a.Animal_Number = c.Animal_Number AND b.Adopter_Number = c.Adopter_Number AND
d.User_Number = c.User_Number AND CAST(c.Adoption_Date AS date) BETWEEN CAST(@start AS
date) AND CAST(@end AS DATE)
ORDER BY a.Animal_Name ASC;
```

### **Total number of adoptions**

```
SELECT COUNT(1)
FROM ADOPTION
WHERE Adoption_Date BETWEEN CAST(@start AS date) AND CAST(@end AS date);
```

### **Animals added per time period**

```
SELECT a.Animal_Number, a.Animal_Name, b.Type_Name, a.Add_Date
FROM ANIMAL a, ANIMAL_TYPE b
```



```
WHERE a.Type_Number = b.Type_Number AND a.Add_Date BETWEEN CAST(@start AS date) AND  
CAST(@end AS date)  
ORDER BY a.Animal_Number ASC;
```

#### **Total number of animals**

```
SELECT COUNT(1)  
FROM ANIMAL  
WHERE Add_Date BETWEEN CAST(@start AS date) AND CAST(@end AS date);
```

#### **All animal type names**

```
SELECT Type_Name  
FROM ANIMAL_TYPE;
```

#### **Number of animals per type**

```
SELECT Type_Name, COUNT(*)  
FROM ANIMAL a, ANIMAL_TYPE b  
WHERE a.Type_Number = b.Type_Number  
GROUP BY Type_Name;
```

#### **Animals added per time period**

```
SELECT Add_Date, COUNT(*)  
FROM ANIMAL GROUP BY Add_Date;
```

### 3. Maintenance Queries (INSERT, UPDATE, DELETE)

#### 3.1 Adopters

```
INSERT INTO ADOPTER (Adopter_Name, Adopter_Surname, Adopter_Contact_Number,  
Adopter_Street_Number, Adopter_Street_Name, City_ID)  
VALUES (@name, @surname, @contactNumber, @streetNumber, @streetName, @cityID);
```

```
UPDATE ADOPTER  
SET Adopter_Name = @name, Adopter_Surname = @surname, Adopter_Contact_Number =  
@contactNumber, Adopter_Street_Number = @streetNumber, Adopter_Street_Name =  
@streetName, City_ID = @cityID  
WHERE Adopter_Number = @adopterNumber;
```

```
DELETE FROM ADOPTER  
WHERE Adopter_Number = @adopterNumber;
```

#### 3.2 Adoptions

```
INSERT INTO ADOPTION (Adopter_Number, Animal_Number, User_Number, Result_Description,  
Adoption_Date)  
VALUES (@adopterNumber, @animalNumber, @userNumber, @resultDescription, @adoptionDate);
```

```
UPDATE ADOPTION  
SET Adopter_Number = @adopterNumber, Animal_Number = @animalNumber, Result_Description  
= @resultDescription  
WHERE Adoption_Number = @adoptionNumber;
```

```
DELETE FROM ADOPTION  
WHERE Adoption_Number = @adoptionNumber;
```

#### 3.3 Animals

```
INSERT INTO ANIMAL (Animal_Name, Type_Number, Add_Date)  
VALUES (@animalName, @animalTypeNumber, @addDate);
```

```
UPDATE ANIMAL  
SET Animal_Name = @newName, Type_Number = @newNumber  
WHERE Animal_Number = @oldNumber;
```

```
DELETE FROM ANIMAL  
WHERE Animal_Name = @name;
```

### 3.4 Animal Types

```
INSERT INTO ANIMAL_TYPE (Type_Name)
VALUES (@typeName);
```

```
UPDATE ANIMAL_TYPE
SET Type_Name = @newTypeName
WHERE Type_Name = @oldTypeName;
```

```
DELETE FROM ANIMAL_TYPE
WHERE Type_Name = @typeName;
```

### 3.5 Cities

```
INSERT INTO CITY (City_Name)
VALUES (@cityName);
```

```
UPDATE CITY
SET City_Name = @cityName
WHERE City_ID = @cityID;
```

```
DELETE FROM CITY
WHERE City_Name = @cityName;
```

### 3.6 Passwords

```
INSERT INTO PASSWORDS (User_Number, Hash, Salt)
VALUES (@userNumber, @hash, @salt);
```

```
UPDATE PASSWORDS
SET Hash = @hash, Salt = @salt
WHERE User_Number = @userNumber;
```

```
DELETE FROM PASSWORDS
WHERE User_Number = @userNumber;
```

### 3.7 Users

```
INSERT INTO USERS (User_Type, User_Name, User_Surname)
VALUES (@typeNumber, @name, @surname);
```

```
UPDATE USERS
SET User_Name = @userName, User_Surname = @userSurname, User_Type = @typeNumber
WHERE User_Number = @userNumber;
```

```
DELETE FROM USERS
WHERE User_Number = @userNumber;
```

### 3.8 User Types

```
INSERT INTO USER_TYPE (Type_Description)  
VALUES (@typeDescription);
```

```
DELETE FROM USER_TYPE  
WHERE Type_Description = @typeDescription;
```