Laboratory 3

Consider 2 tables:

Person	**Order**
#Pid int	#Oid int
FirstName	Pid
LastName	
City	

1. Add new column

```
ALTER TABLE Person
ADD Dob date
```

2. Modify the type of a column

```
ALTER TABLE Person
ALTER COLUMN Dob int NOT NULL
```

3. Remove a column

```
ALTER TABLE Person
DROP COLUMN Dob
```

4. Create new table

```
CREATE TABLE Person(
Pidint NOT NULL PRIMARY KEY,
FirstName varchar(50) NOT NULL,
LastName varchar(50),
City varchar(50)
);
```

5. Add new column with default constraint

```
ALTER TABLE Person
ADD  Dob int DEFAULT 2000;
```

6. Modify column with default constraint

```
ALTER TABLE Person
ADD DEFAULT 18 FOR Age;

ALTER TABLE Person
ADD CONSTRAINT df_18 DEFAULT 18
FOR Age
```

7. Remove default constraint from a column

```
ALTER TABLE Person
DROP CONSTRAINT df_18;
```

8. Delete a table
   - Delete all the structure of the table and the records

     ```
     DROP TABLE Person
     ```

   - Delete only the records (with condition)

     ```
     DELETE FROM Person
     [WHERE Dob>2000]
     ```

9. Create a foreign key constraint on a new table

```
CREATE TABLE Order (
Oid int NOT NULL PRIMARY KEY,
Pid int CONSTRAINT fk_Order_Person FOREIGN KEY(Pid) REFERENCES Person(Pid)
);
```

10. Create a foreign key as a new add column in a table

```
ALTER TABLE Order
ADD CONSTRAINT fk_Order_Person FOREIGN KEY(Pid) REFERENCES Person(Pid)
```

11. Remove a foreign key

```
ALTER TABLE Order
DROP CONSTRAINT fk_Order_Person;
```

12. Create a primary key constraint in a new table

```
CREATE TABLE Order (
Oid INT NOT NULL,
Pid INT CONSTRAINT fk_Order_Person FOREIGN KEY(Pid) REFERENCES Person(Pid)
CONSTRAINT pk_Order PRIMARY KEY(Oid)
);
```

13. Create a primary key constraint as a new add column in a table already created

```
ALTER TABLE Order
ADD CONSTRAINT pk_Order PRIMARY KEY(Oid)
```

14. Remove a primary key constraint

```
ALTER TABLE Order
DROP CONSTRAINT pk_Order
```

15. Create a unique constraint in a new table (secondary key)

```
CREATE TABLE Order(
Oid INT NOT NULL
CONSTRAINT uk_Order UNIQUE(Oid)
);
```

16. Create a unique constraint as a new add column in a table

```
ALTER TABLE Order
ADD CONSTRAINT uk_Order UNIQUE(Oid)
```

17. Remove a unique constraint

```
ALTER TABLE Order
DROP CONSTRAINT uk_Order
```

7 procedures do	**7 procedures undo (reverse)**
do_proc_1 – modify the type of the column	undo_proc_1 – modify the type of the column (back)
do_proc_2 – add a column	undo_proc_2 – remove a column
do_proc_3 – add a default constraint	undo_proc_3 – remove a default constraint
do_proc_4 – create a primary key	undo_proc_4 – remove a primary key
do_proc_5 – create a secondary key (unique)	undo_proc_5 – remove a secondary key (unique)
do_proc_6 – create a foreign key constraint	undo_proc_6 – remove a foreign key constraint
do_proc_7 – create a table	undo_proc_7 – remove a table

PAY ATTENTION to the name of the procedures – because with their names you work in the main procedure (but please don't use the ones from up).
A table Version will contains the version of the database (version 0 – the first one – the one it is now)

main 4 – will take the database from version 0 to version 4 (crossing version 1, 2, 3)
version 1 – will be given by executing do_proc_1
version 2 – will be given by executing do_proc_2
version 3 – will be given by executing do_proc_3

main 2 – will take the database from version 4 (the one you have now) to version 2 (crossing version 3)
version 3 – will be given by executing undo_proc_3

STORED PROCEDURES can be found in the Database (your database) -> Programmability -> Stored Procedures -> (right click) Stored Procedures.

Examples of a stored procedure name without parameter:

```
CREATE PROCEDURE do_proc_1
AS
BEGIN
   -- the code
  SELECT * FROM Produs
END
/* EXECUTE (to create the stored procedure and find it in the list of stored procedures */
```

```
Run the procedure (in a new query):
EXECUTE do_proc_1 /
EXEC do_proc_1 /
do_proc_1
```

Examples of a stored procedure with parameters:

```
CREATE PROCEDURE main
@vers int,
@t varchar(50)
AS
BEGIN
 IF @vers>5
 BEGIN
   SELECT * FROM Produs
 END
 IF @t='Cluj'
 BEGIN
   PRINT ' DONE'
 END
END
```

```
Run the procedure (in a new query):
EXEC main 6, 'Alba' /
EXEC main 1, 'Cluj' /
EXEC main 7, 'Cluj'
```

Each stored procedure will have a different name and after EXECUTE it will appear in the list of the stored procedures (at Refresh). This means that the procedure was created and can be used (in main procedure or wherever you want).

To run the procedure: open a New Query and write EXECUTE procedure_name [parameters].

EXECUTE main 3 / EXEC main 4 / EXEC main @vers=3

Instructions:

```
1. WHILE condition
   BEGIN
   ….
   END
```

```
2. IF condition
   BEGIN
   …
   END
   [ELSE
   BEGIN
   …
   END]
```

3. PRINT 'Your message.';
4. DECLARE @a INT -- to declare a variable
5. SET @a=@a +1 -- to modify the value of a variable
6. EXEC @text -- to execute (run) the instruction saved in variable @text