Бази Даних

Лабораторні роботи 1-3

ФІ-91, Корешков Михайло

Варіант 7

# Лабораторна робота 1

Спочатку створюю всі таблиці. Попередньо видаляю їх, щоб результат виконання скрипта був однаковим незалежно від стану БД.

[file: lab1.sql]

1. use lab1;
2. drop table Abonents;
3. drop table Rooms;
4. drop table Divisions;
5. GO
6. create table Divisions (
7. DivisionName varchar(32) primary key,
8. DivisionType varchar(32)
9. );
10. create table Rooms (
11. RoomNumber varchar(10) primary key,
12. RoomName varchar(32) null,
13. RoomType varchar(16) null,
14. RoomFloor SMALLINT,
15. DivisionName varchar(32)
16. );
17. create table Abonents (
18. Number varchar(16) primary key,
19. PName varchar(32) not null,
20. IName varchar(32) not null,
21. BName varchar(32) not null,
22. DOB varchar(10) not null,
23. RoomNumber varchar(10),
24. DivisionName varchar(32),
25. );

Output:

Commands completed successfully.

Тепер заповнюю даними таблиці Divisions та Rooms - щоб далі комфортно тестити роботу зовнішніх ключів.

[file: lab1\_1\_insert\_divisions.sql]

1. USE [lab1]
2. GO
3. DELETE FROM [dbo].[Divisions];
4. GO
5. INSERT INTO [dbo].[Divisions]
6. ([DivisionName]
7. ,[DivisionType])
8. VALUES
9. ('Reception', 'Type1'),
10. ('Security', 'Type1'),
11. ('Engineering', 'Type2'),
12. ('HR', 'Type3'),
13. ('IT', 'Type3'),
14. ('Admin', 'Type3'),
15. ('Accounting', 'Type3')
16. GO

Output:

(0 rows affected)

(7 rows affected)

[file: lab1\_2\_insert\_rooms.sql]

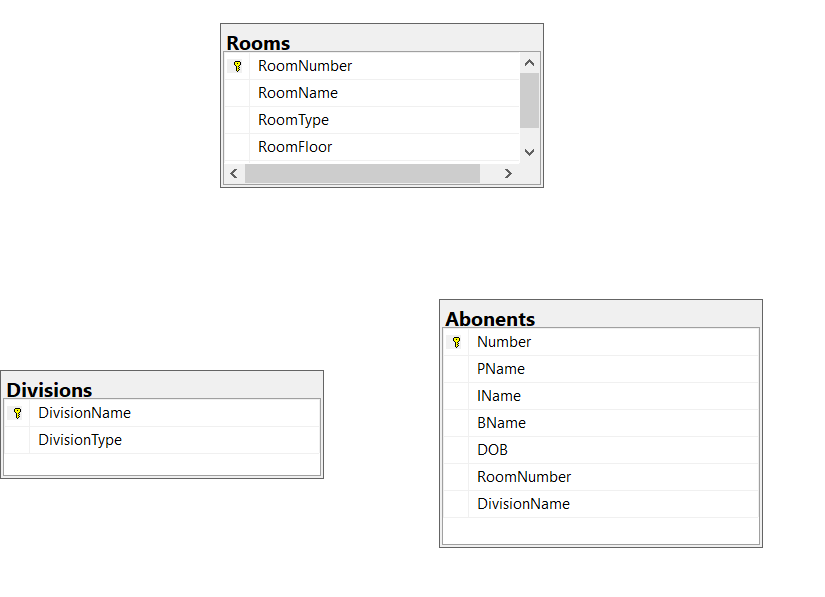
1. USE [lab1]
2. GO
3. DELETE FROM [dbo].[Rooms];
4. GO
5. INSERT INTO [dbo].[Rooms]
6. ([RoomNumber]
7. ,[RoomName]
8. ,[RoomType]
9. ,[DivisionName]
10. ,[RoomFloor])
11. VALUES
12. ('001' , 'Reception', null, 'Reception', 1),
13. ('002' , 'Outdoor Security', null, 'Security', 1),
14. ('101' , 'Security HQ', 'Room', 'Security', 1),
15. ('102' , 'Client Services', 'Room', 'Reception', 1),
16. ('103' , 'Client Manager', 'Room', 'Reception', 1),
17. ('201' , 'Administration', 'Room', 'Admin', 2),
18. ('201-2', 'CEO', 'Room', 'Admin', 2),
19. ('202' , 'IT', 'Room', 'IT', 2),
20. ('202-2', 'CTO', 'Room', 'IT', 2),
21. ('210' , 'IT Open Space', 'Open Space', 'IT', 2),
22. ('210-1', null, 'Cubicle', 'IT', 2),
23. ('210-2', null, 'Cubicle', 'IT', 2),
24. ('210-3', null, 'Cubicle', 'IT', 2),
25. ('210-4', null, 'Cubicle', 'IT', 2),
26. ('210-5', null, 'Cubicle', 'IT', 2),
27. ('210-6', null, 'Cubicle', 'IT', 2),
28. ('210-7', null, 'Cubicle', 'IT', 2),
29. ('210-8', null, 'Cubicle', 'IT', 2),
30. ('220', 'Engineer On-Duty', 'Room', 'Engineering',2 ),
31. ('221', 'Engineering', 'Open Space', 'Engineering',2),
32. ('221-1', null, 'Desk', 'Engineering', 2),
33. ('221-2', null, 'Desk', 'Engineering', 2),
34. ('221-3', null, 'Desk', 'Engineering', 2),
35. ('221-4', null, 'Desk', 'Engineering', 2),
36. ('330', 'Chief Accountant', 'Room', 'Admin', 3),
37. ('331', 'Finances', null, 'Admin', 3),
38. ('331-1', null, 'Desk', 'Admin', 3),
39. ('331-2', null, 'Desk', 'Admin', 3),
40. ('331-3', null, 'Desk', 'Admin', 3),
41. ('331-4', null, 'Desk', 'Admin', 3)
42. GO

Output:

(0 rows affected)

(30 rows affected)

## Діаграма 1. До створення зовнішніх ключів:



# Лабораторна робота 2

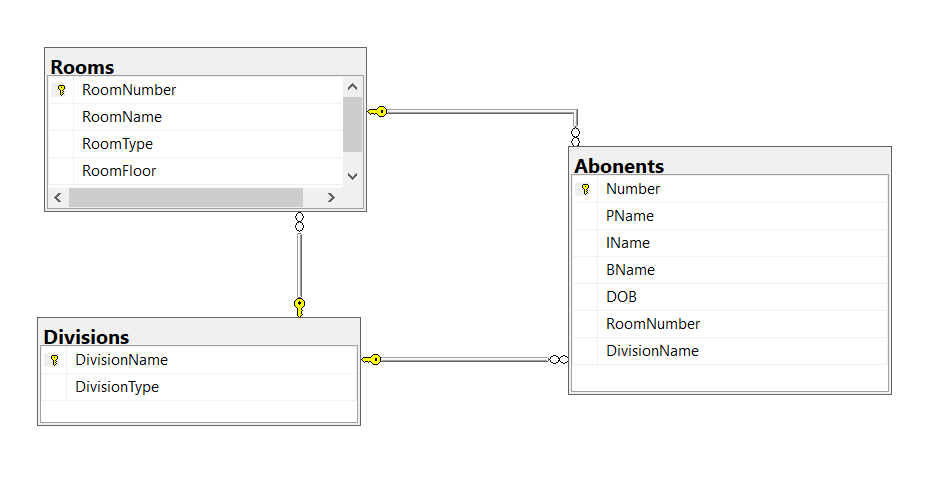
Тепер додамо зовнішні ключі та обмеження:

[file: lab2\_1\_add\_foreignkeys.sql]

1. use lab1;
2. go
3. alter table Rooms
4. add constraint
5. RoomsDivisionForeign foreign key (DivisionName)
6. references Divisions;
7. alter table Abonents add
8. constraint
9. AbonentRoomForeign foreign key (RoomNumber)
10. references Rooms
11. on delete cascade,
12. constraint
13. AbonentDivisionForeign foreign key (DivisionName)
14. references Divisions
15. on delete cascade;
16. GO

Commands completed successfully.

## Діаграма 2: БД після додавання зовнішніх ключів



Тепер цікавіше – граємось з nocheck.

[file: lab2\_2\_nocheck.sql]

1. use lab1;
2. go
3. delete from Abonents;
4. go
5. ALTER TABLE Abonents
6. check constraint AbonentDivisionForeign;
7. go
8. INSERT INTO [dbo].[Abonents]
9. VALUES
10. ('111-11-11', 'Lastname1', 'Name1', 'Fathername1', '01-11-2001', '101', 'Security'),
11. ('111-11-12', 'Lastname2', 'Name2', 'Fathername2', '02-11-2001', '101', 'UNNAMED DIVISION');
12. GO
13. -- ^ error. The INSERT statement conflicted with the FOREIGN KEY constraint "AbonentDivisionForeign"
14. ALTER TABLE Abonents
15. nocheck constraint AbonentDivisionForeign;
16. INSERT INTO [dbo].[Abonents]
17. VALUES
18. ('111-11-11', 'Lastname1', 'Name1', 'Fathername1', '01-11-2001', '101', 'Security'),
19. ('111-11-12', 'Lastname2', 'Name2', 'Fathername2', '02-11-2001', '101', 'UNNAMED DIVISION');
20. GO
21. select \* from Abonents;
22. go
23. -- ^ everything ok.
24. ALTER TABLE Abonents
25. check constraint AbonentDivisionForeign;
26. go
27. -- ^ ok
28. INSERT INTO [dbo].[Abonents]
29. VALUES
30. ('111-11-13', 'Lastname3', 'Name3', 'Fathername3', '03-11-2001', '101', 'UNNAMED DIVISION 2');
31. GO
32. -- ^ error. The INSERT statement conflicted with the FOREIGN KEY constraint "AbonentDivisionForeign"
33. ALTER TABLE Abonents
34. with check
35. check constraint AbonentDivisionForeign;
36. go
37. -- ^ error. now we are checking old data.
38. -- ALTER TABLE statement conflicted with the FOREIGN KEY constraint "AbonentDivisionForeign"
39. select \* from
40. Abonents A left join Divisions D
41. on A.DivisionName = D.DivisionName
42. where D.DivisionName is null;
43. -- ^ here are all Abonents with invalid DivisionName
44. -- Now lets delete all Abonents with invalid DivisionName
45. delete A from
46. Abonents A left join Divisions D
47. on A.DivisionName = D.DivisionName
48. where D.DivisionName is null;
49. -- Done!
50. ALTER TABLE Abonents
51. with check
52. check constraint AbonentDivisionForeign;
53. go
54. -- ^ ok!
55. INSERT INTO [dbo].[Abonents]
56. VALUES
57. ('111-11-13', 'Lastname3', 'Name3', 'Fathername3', '03-11-2001', '101', 'UNNAMED DIVISION 2');
58. GO
59. -- ok, error!

Output:

(1 row affected)

Msg 547, Level 16, State 0, Line 10

The INSERT statement conflicted with the FOREIGN KEY constraint "AbonentDivisionForeign". The conflict occurred in database "lab1", table "dbo.Divisions", column 'DivisionName'.

The statement has been terminated.

(2 rows affected)

Number PName IName BName DOB RoomNumber DivisionName

----------- ------------- -------- -------------- ---------- ---------- --------------

111-11-11 Lastname1 Name1 Fathername1 01-11-2001 101 Security

111-11-12 Lastname2 Name2 Fathername2 02-11-2001 101 UNNAMED DIVISION

(2 rows affected)

Msg 547, Level 16, State 0, Line 38

The INSERT statement conflicted with the FOREIGN KEY constraint "AbonentDivisionForeign". The conflict occurred in database "lab1", table "dbo.Divisions", column 'DivisionName'.

The statement has been terminated.

Msg 547, Level 16, State 0, Line 46

The ALTER TABLE statement conflicted with the FOREIGN KEY constraint "AbonentDivisionForeign". The conflict occurred in database "lab1", table "dbo.Divisions", column 'DivisionName'.

Number PName IName BName DOB RoomNumber DivisionName DivisionName DivisionType

---------- ------------ ------- ------------ ---------- ---------- ------------------ ------------- ------------

111-11-12 Lastname2 Name2 Fathername2 02-11-2001 101 UNNAMED DIVISION NULL NULL

(1 row affected)

(1 row affected)

Msg 547, Level 16, State 0, Line 74

The INSERT statement conflicted with the FOREIGN KEY constraint "AbonentDivisionForeign". The conflict occurred in database "lab1", table "dbo.Divisions", column 'DivisionName'.

The statement has been terminated.

Completion time: 2021-10-11T19:03:17.0741747+03:00

Отже:

* Зовнішній ключ не дає додати абонента з неправильним підрозділом
* Відключаємо цей ключ
  + Тепер можемо додати, додаємо
* Підключаємо знову цей ключ
  + **Без проблем,** бо не написали явно «**with check**» - неправильні записи залишились
* Підключаємо знову, але вже з **with check**
  + Помилка, як і очікувалось. Потрібно спочатку видалити невалідних абонентів
* Видаляємо невалідних абонентів за допомогою конструкції **delete … left join …**
  + Вибираємо всіх Абонентів та відповідні їх Підрозділи
  + Шукаємо всі записи, де абоненту не відповідає ніякий підрозділ, тобто **DivisionName is null**
  + Видаляємо ці записи
* Підключаємо знову з **with check**
  + Тепер все ок
* Пробуємо додати невалідного абонента
  + Помилка, як і очікувалось

[file: lab2\_3\_alter\_column.sql]

1. use lab1;
2. go
3. drop table Abonents;
4. go
5. create table Abonents (
6. Number varchar(16) primary key,
7. PName varchar(32) not null,
8. IName varchar(32) not null,
9. BName varchar(32) not null,
10. DOB varchar(10) not null,
11. RoomNumber varchar(10),
12. DivisionName varchar(32),
13. );
14. go
15. INSERT INTO [dbo].[Abonents]
16. VALUES
17. ('111-11-11', 'Lastname1', 'Name1', 'Fathername1', '01-11-2001', '101', 'Security'),
18. ('111-11-12', 'Lastname2', 'Name2', 'Fathername2', '01-11-2001', '101', 'Security')
19. GO
20. alter table Abonents
21. add Active char(1) not null
22. constraint AbonentActiveDefault default 'Y';
23. go
24. select \* from Abonents;
25. alter table Abonents
26. drop column Active;
27. go
28. -- error because "The object 'AbonentActiveDefault' is dependent on column 'Active'."
29. alter table Abonents
30. drop constraint AbonentActiveDefault;
31. go
32. alter table Abonents
33. drop column Active;
34. go
35. ----------------
36. alter table Rooms
37. with check
38. add constraint RoomFloorCheck check (RoomFloor between 1 and 3);
39. go
41. select \* from Rooms where RoomType='Room';
42. -- Let's add new single room on the roof - 4th floor.
43. -- We need to temporarly disable constraint RoomFloorCheck
44. alter table Rooms
45. nocheck constraint RoomFloorCheck;
46. go
47. insert Rooms values ('404', 'Roof Birdwathing Room', 'Room', 4, 'Admin')
48. -- turn it back on
49. alter table Rooms
50. with nocheck
51. check constraint RoomFloorCheck;
52. go
53. select \* from Rooms where RoomFloor >= 3;
54. ----------------
55. select \* from Abonents;
56. -- ^ ok
57. exec sp\_rename 'Abonents', 'Abonenty'
58. go
59. select \* from Abonenty;
60. -- ^ ok
61. exec sp\_rename 'Abonenty', 'Abonents'
62. go

Output:

(2 rows affected)

(2 rows affected)

Msg 5074, Level 16, State 1, Line 30

The object 'AbonentActiveDefault' is dependent on column 'Active'.

Msg 4922, Level 16, State 9, Line 30

ALTER TABLE DROP COLUMN Active failed because one or more objects access this column.

(9 rows affected)

(1 row affected)

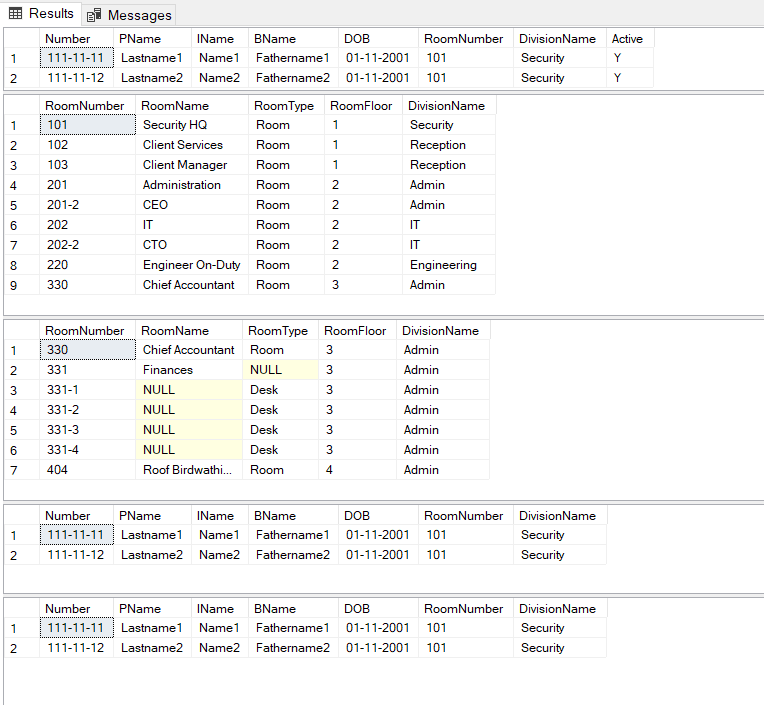
(7 rows affected)

(2 rows affected)

Caution: Changing any part of an object name could break scripts and stored procedures.

(2 rows affected)

Caution: Changing any part of an object name could break scripts and stored procedures.



Що тут діється:

Частина 1 – новий дефолтний атрибут

* Відновлюємо стан таблиць
* Додаємо декілька Абонентів
* Модифікуємо таблицю Абонентів так, щоб додати новий атрибут **«Активний»**
  + За замовчуванням атрибут дорівнює «Y»
  + Оскільки атрибут **not null**, це дефолтне значення автоматично надано всім існуючим записам таблиці. Інакше для цього потрібно було б дописати «with values»
  + Бачимо, що поле додано в першому селекті
* Намагаємось видалити атрибут
  + Помилка, бо на атрибуті висить обмеження «дефолт».
* Знімаємо обмеження, знову намагаємось видалити атрибут
  + Тепер все ок

Частина 2 – нове обмеження та його тимчасове вимкнення

* Додаємо обмеження **check**, що перевірятиме, чи нормальний поверх має Кімната.
  + Я припустив, що в будівлі 3 поверхи.
* Згодом ми дізналися, що є ще одна кімната на даху – 4-му поверсі. Треба її додати також
  + Це виключення
  + Тимчасово вимикаємо обмеження
  + Додаємо кімнату
  + Вмикаємо обмеження назад із **with** **nocheck**
* Бачимо, що кімната залишається в таблиці

Частина 3 – Перейменування таблиці

* Змінимо назву таблиці за допомогою команди exec sp\_rename
* І те саме назад.
* Кожен раз БД нас попереджає, що це несе непередбачувані наслідки, бо може бути багато об’єктів, що залежать від старої назви

# Лабораторна робота 3

Заповнення таблиць даними

Вже заповнено раніше.