

Міністерство освіти і науки України

НТУУ«Київський політехнічний інститут»

Фізико-технічний інститут

Розрахунково-графічна робота

з дисципліни

«Інформаційно-комунікаційні системи 1.

Бази даних та ІКС»

Варіант №10

**Виконав:**

Студент III курсу ФТІ

групи ФБ-31

Костенко Максим

**Перевірив:**

Коломицев М.В.

Київ 2015

1. Завдання для розрахунково-графічної роботи**:**

1.1. Промоделювати функціональні вимоги до системи, що проектується за допомогою DFD-діаграми, для чого:

* Ідентифікувати зовнішні об'єкти, що є постачальниками / приймачами інформації, і види інформації, які циркулюють між системою і зовнішніми об'єктами;
* Визначити основні функціональні вимоги до системи і представити їх у вигляді процесів;
* Визначити основні види інформації і представити їх у вигляді потоків між процесами і зовнішніми об'єктами;
* Задати сховища інформації.

Для побудови DFD-діаграми використовувати інструментальний засіб CASE Studio2.

1.2. Побудувати концептуальну модель (ER-модель), для чого:

* За описом предметної області і поставленому завданню виявити необхідний набір сутностей, їх властивості і зв'язки між сутностями;
* Визначити необхідний набір атрибутів для кожної сутності, виділивши атрибути що ідентифікують сутність;
* Класифікувати сутності (стрижневі, асоціативні та інші);
* Визначити зв'язки між об'єктами;
* Задати обмеження цілісності;
* Описати отриману ER-модель.

Для побудови ER-моделі використовувати інструментальний засіб CASE Studio 2.

1.3. Отримати реляційну схему з ER-моделі, для чого:

* Побудувати набір необхідних таблиць бази даних;
* Виділити первинні та зовнішні ключі певних таблиць;
* Привести отримані таблиці до третьої нормальної форми.
* Дати математичний опис отриманої моделі з використанням апарату реляційної алгебри.

1.4. Використовуючи СУБД MS SQL Server створити спроектовану базу даних.

1.5. На мові SQL записати вирази для зазначених у варіанті завдання типів запитів. Перевірити працездатність написаних запитів в інтерактивному режимі.

# 2. Особисте завдання: Варіант №10

**Предметна область** – готель.

**Задачі, що вирішує інформаційна система** – облік готельного фонду.

**Завдання на обробку даних:**

1. Створити впорядковані списки:

- Номерів (місць), вивільнюваних сьогодні і завтра;

- Порожніх номерів;

- Номерів, в яких немає вільних місць;

- Номерів, в яких є вільні місця.

2. обчислення:

- Перевірити, чи є в наявності вільні номери для жінок (чоловіків).

3. корекція:

- Зміна номера, в якому зупинився постоялець,

- Зміна дати виселення.

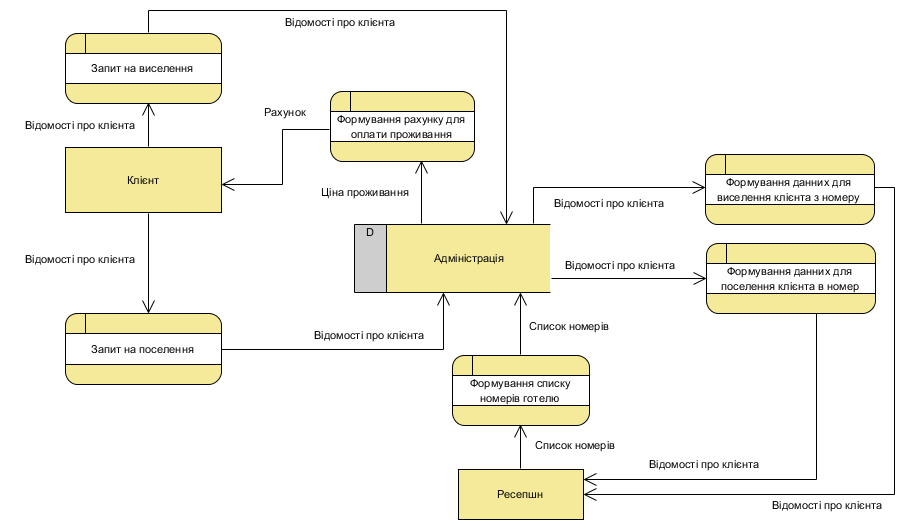
4. Звіти виду:

- "Постояльці, які проживають у готелі на даний час".

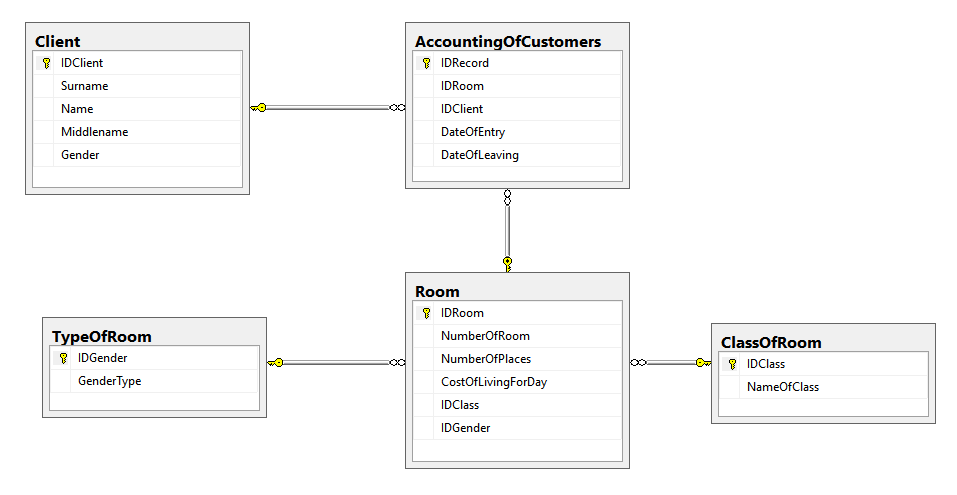
- "Вільні місця": клас - номер - загальна кількість місць в номері - кількість вільних місць.

- "Рахунок на оплату номера" (запит для конкретного постояльця).

# 3. Побудова DFD-діаграми



# 4. Побудова ER-діаграми



# 5. Нормалізація

1. TypeOfRoom(IDGender,GenderType)

IDGender → GenderType

Всі атрибути атомарні та немає повторюваних груп тому відношення TypeOfRoom належить до 1НФ. Всі неключові атрибути функціонально залежні від цілих потенційних ключів тому TypeOfRoom належить до 2НФ. Також немає транзитивних залежностей тому TypeOfRoom належить до 3НФ.

2. ClassOfRoom(IDClass, NameOfClass)

IDClass→ NameOfClass

Всі атрибути атомарні та немає повторюваних груп тому відношення ClassOfRoom належить до 1НФ. Всі неключові атрибути функціонально залежні від цілих потенційних ключів тому ClassOfRoom належить до 2НФ. Також немає транзитивних залежностей тому ClassOfRoom належить до 3НФ.

3. Client(IDClient,Surname,Name, Middlename,Gender)

IDClient → Surname; IDClient → Name; IDClient → Middlename; IDClient → Gender

Всі атрибути атомарні та немає повторюваних груп тому відношення Client належить до 1НФ. Всі неключові атрибути функціонально залежні від цілих потенційних ключів тому Client належить до 2НФ. Також немає транзитивних залежностей тому Client належить до 3НФ.

4.Room(IDRoom,NumberOfRoom,NumberOfPlaces,CostOfLivingForDay,IDClass,

IDGender)

IDRoom → NumberOfRoom; IDRoom → NumberOfPlaces;

IDRoom → CostOfLivingForDay; IDRoom → IDClass; IDRoom → IDGender

Всі атрибути атомарні та немає повторюваних груп тому відношення Room належить до 1НФ. Всі неключові атрибути функціонально залежні від цілих потенційних ключів тому Room належить до 2НФ. Також немає транзитивних залежностей тому Room належить до 3НФ.

5. AccountingOfCustomers(IDRecord,IDRoom,IDClient,DateOfEntry,DateOfLeaving)

IDRecord → IDRoom; IDRecord → IDClient; IDRecord → DateOfEntry;

IDRecord → DateOfLeaving

Всі атрибути атомарні та немає повторюваних груп тому відношення AccountingOfCustomers належить до 1НФ. Всі неключові атрибути функціонально залежні від цілих потенційних ключів тому AccountingOfCustomers належить до 2НФ. Також немає транзитивних залежностей тому AccountingOfCustomers належить до 3НФ.

# 

# 6. Завдання на обробку даних

Створення бази даних та обмежень за допомогою SQL-команд:

CREATE TABLE TypeOfRoom

(IDGender INT IDENTITY CONSTRAINT GenderPrimary PRIMARY KEY,

GenderType VARCHAR(10))

CREATE TABLE ClassOfRoom

(IDClass INT IDENTITY CONSTRAINT ClassPrimary PRIMARY KEY,

NameOfClass VARCHAR(10))

CREATE TABLE Client

(IDClient INT IDENTITY CONSTRAINT ClientPrimary PRIMARY KEY,

Surname VARCHAR(20) NOT NULL

CONSTRAINT SurnameStudentCheck CHECK (Surname NOT LIKE '%[^a-z][0-9]%'),

Name VARCHAR(20) NOT NULL

CONSTRAINT NameStudentCheck CHECK (Name NOT LIKE '%[^a-z][0-9]%'),

Middlename VARCHAR(20) NOT NULL

CONSTRAINT MiddlenameStudentCheck CHECK (Middlename NOT LIKE '%[^a-z][0-9]%'))

CREATE TABLE Room

(IDRoom INT IDENTITY CONSTRAINT RoomPrimary PRIMARY KEY,

NumberOfRoom INT,

NumberOfPlaces INT,

CostOfLivingForDay MONEY,

IDClass INT,

CONSTRAINT RoomClass FOREIGN KEY (IDClass) REFERENCES ClassOfRoom,

IDGender INT,

CONSTRAINT RoomForSomeGenderType FOREIGN KEY (IDGender) REFERENCES TypeOfRoom)

CREATE TABLE AccountingOfCustomers

(IDRecord INT IDENTITY CONSTRAINT RecordPrimary PRIMARY KEY,

IDRoom INT,

CONSTRAINT RoomOfClient FOREIGN KEY (IDRoom) REFERENCES Room,

IDClient INT,

CONSTRAINT Clients FOREIGN KEY (IDClient) REFERENCES Client,

DateOfEntry date,

DateOfLeaving date,

CostOfLiving MONEY)

Проаналізувавши сутності системи, я визначив, які обмеження на дані потрібно вводити та створив тригери на вставку недопустимих кортежів, для забезпечення цілісності даних в таблиці AccountingOfCustomers:

1)В кімнату не може бути поселено клієнтів більше за загальну кількість місць в цій кімнаті:

USE Hotel;

GO

CREATE TRIGGER ADDRecord

ON AccountingOfCustomers FOR INSERT

AS

IF EXISTS (SELECT R.NumberOfRoom,COUNT (IDClient) QuantityOfClients

FROM Room R,AccountingOfCustomers A

WHERE A.IDRoom = R.IDRoom

GROUP BY R.NumberOfRoom,NumberOfPlaces

HAVING COUNT (IDClient) > NumberOfPlaces)

BEGIN

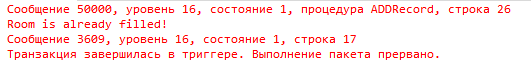
RAISERROR('Room is already filled!',16,1)

ROLLBACK TRAN

END

INSERT INTO AccountingOfCustomers(IDRoom,IDClient,DateOfEntry,DateOfLeaving)

VALUES(1,5,'25.12.15','29.12.15')



2)Клієнт може бути поселеним лише в кімнату, тип якої відповідає статі клієнта:

CREATE TRIGGER CheckGender

ON AccountingOfCustomers FOR INSERT

AS

IF EXISTS (SELECT Surname,Name, Middlename

FROM Room R,AccountingOfCustomers A,TypeOfRoom T,Client C

WHERE A.IDRoom = R.IDRoom AND A.IDClient = C.IDClient AND T.IDGender = R.IDGender AND C.Gender != T.GenderType)

BEGIN

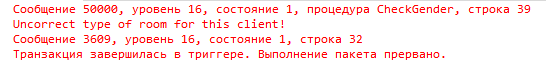
RAISERROR('Uncorrect type of room for this client!',16,1)

ROLLBACK TRAN

END

INSERT INTO AccountingOfCustomers(IDRoom,IDClient,DateOfEntry,DateOfLeaving)

VALUES(2,7,'25.12.15','29.12.15')



1. Створити впорядковані списки:

- Номерів (місць), вивільнюваних сьогодні і завтра;

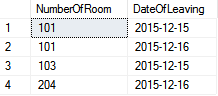
USE HOTEL;

SELECT NumberOfRoom,DateOfLeaving FROM Room R,AccountingOfCustomers A

WHERE R.IDRoom = A.IDRoom AND (DateOfLeaving = (SELECT CONVERT (DATE, GETDATE()))

OR DateOfLeaving=(SELECT DATEADD(DAY, DATEDIFF(DAY, '20000101', CURRENT\_TIMESTAMP), '20000102')))

GROUP BY NumberOfRoom,DateOfLeaving



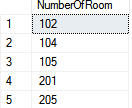
- Порожніх номерів;

SELECT R.NumberOfRoom FROM ROOM R

EXCEPT

SELECT R.NumberOfRoom

FROM Room R,AccountingOfCustomers A WHERE A.IDRoom = R.IDRoom



- Номерів, в яких немає вільних місць;

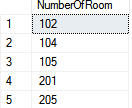
SELECT R.NumberOfRoom,COUNT (IDClient) QuantityOfClients

FROM Room R,AccountingOfCustomers A

WHERE A.IDRoom = R.IDRoom

GROUP BY R.NumberOfRoom,NumberOfPlaces

HAVING COUNT (IDClient) = NumberOfPlaces



- Номерів, в яких є вільні місця.

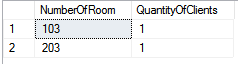
SELECT R.NumberOfRoom,COUNT (IDClient) QuantityOfClients

FROM Room R,AccountingOfCustomers A

WHERE A.IDRoom = R.IDRoom

GROUP BY R.NumberOfRoom,NumberOfPlaces

HAVING COUNT (IDClient) < NumberOfPlaces



2. обчислення:

- Перевірити, чи є в наявності вільні номери для жінок (чоловіків).

SELECT R.NumberOfRoom FROM ROOM R,TypeOfRoom T

WHERE R.IDGender = T.IDGender AND GenderType = 'Female'

EXCEPT

SELECT R.NumberOfRoom

FROM Room R, TypeOfRoom T,AccountingOfCustomers A

WHERE R.IDGender = T.IDGender AND GenderType = 'Female' AND A.IDRoom = R.IDRoom

GROUP BY R.NumberOfRoom



3. Корекція:

- Зміна номера, в якому зупинився постоялець

CREATE TRIGGER UpdateRoom

ON AccountingOfCustomers

FOR UPDATE

AS

IF EXISTS (SELECT I.IDRoom,D.IDRoom FROM INSERTED I , DELETED D WHERE I.IDRoom = D.IDRoom OR I.IDRoom < 1)

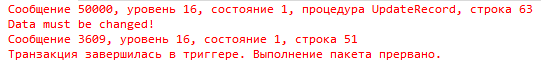
BEGIN

RAISERROR('Data must be changed!',16,1)

ROLLBACK TRAN

END

UPDATE AccountingOfCustomers SET IDRoom = 9 WHERE IDRoom = 9



- Зміна дати виселення.

CREATE TRIGGER UpdateDate

ON AccountingOfCustomers

FOR UPDATE

AS

IF EXISTS (SELECT I.DateOfLeaving,D.DateOfLeaving FROM INSERTED I , DELETED D

WHERE I.DateOfLeaving = D.DateOfLeaving OR I.DateOfLeaving < GETDATE())

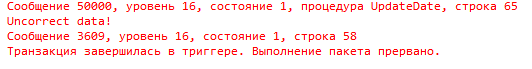
BEGIN

RAISERROR('Uncorrect data!',16,1)

ROLLBACK TRAN

END

UPDATE AccountingOfCustomers SET DateOfLeaving ='16.12.15' WHERE DateOfLeaving ='16.12.15'



4. Звіти виду:

- "Постояльці, які проживають у готелі на даний час".

CREATE PROCEDURE LivingNow

AS

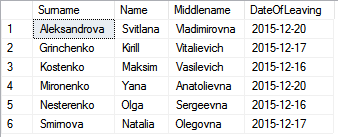
SELECT Surname,Name, Middlename,DateOfLeaving

FROM Client C,AccountingOfCustomers A

WHERE C.IDClient=A.IDClient AND DateOfEntry <= GETDATE() AND DateOfLeaving > GETDATE()

GROUP BY Surname,Name, Middlename,DateOfLeaving

EXEC LivingNow



- "Вільні місця": клас - номер - загальна кількість місць в номері - кількість вільних місць.

CREATE PROCEDURE FreePlaces

AS

SELECT NameOfClass,NumberOfRoom,NumberOfPlaces, NumberOfPlaces - COUNT (IDClient) NumberOfFreePlaces

FROM Room R LEFT JOIN AccountingOfCustomers A ON A.IDRoom = R.IDRoom

INNER JOIN ClassOfRoom C ON R.IDClass = C.IDClass

WHERE (SELECT COUNT (A1.IDClient) QuantityOfClients1

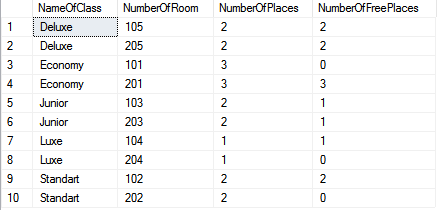
FROM Room R1,AccountingOfCustomers A1

WHERE A1.IDRoom = R1.IDRoom) IS NOT NULL

GROUP BY NameOfClass,NumberOfRoom,NumberOfPlaces

ORDER BY NameOfClass,NumberOfRoom

EXEC FreePlaces



- "Рахунок на оплату номера" (запит для конкретного постояльця).

CREATE PROCEDURE CostOfLeaving

AS

SELECT Surname,Name, Middlename,(DATEDIFF ( DAY , DateOfEntry , DateOfLeaving ))\*CostOfLivingForDay AS 'CostOfLeaving'

FROM Client C,AccountingOfCustomers A ,Room R

WHERE C.IDClient=A.IDClient AND R.IDRoom = A.IDRoom

EXEC CostOfLeaving

