

Міністерство освіти і науки України

Харківський національний університет імені В.Н. Каразіна
Кафедра теоретичної та прикладної інформатики

Звіт по дисципліні
Вступ до SQL баз даних

Індивідуальне завдання № 5

Студента: Чистякова Артема
Групи: МФ-31

Необхідний термін здачі завдання: 19.11.2020

Фактичний термін здачі завдання: _____

Кількість балів: _____

Харків 2020

Перечень запросов (общее задание)

1. Создать две таблицы Person и Country. Заполнить их указанными значениями.

```
CREATE TABLE Person
(
    id_person INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
    Name VARCHAR(50),
    ref_country INT NULL
);

CREATE TABLE Country
(
    id_country INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
    name_country VARCHAR(50)
);

ALTER TABLE Person
ADD CONSTRAINT constr_Person_Country
FOREIGN KEY (ref_country)
REFERENCES Country (id_country);
```

2. Вывести всех людей и названия стран, в которых они живут (использовать инструкцию inner join внутреннего соединения).

```
SELECT Name, name_country FROM Person
INNER JOIN Country ON ref_country = id_country;
```

3. Вывести всех людей и страны в которых они живут, а также тех, у кого не зарегистрирована страна (использовать внешнее объединение left outer join).

```
SELECT Name, name_country FROM Person
LEFT JOIN Country ON ref_country = id_country;
```

4. Вывести список стран, в которых не зарегистрирован ни один человек (использовать правое внешнее объединение right join).

```
SELECT name_country FROM Person
RIGHT JOIN Country ON ref_country = id_country
WHERE ref_country IS NULL;
```

5. Вывести список людей, у которых страна проживания – Украина.

```
SELECT Name FROM Person
INNER JOIN Country ON ref_country = id_country
WHERE name_country = "Ukraine";
```

6. Создать две таблицы Firm и Telephone. Заполнить их указанными значениями.

```
CREATE TABLE Firm
(
    id_firm INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
    name_firm VARCHAR(50),
    ref_country INT NULL
);
```

```
CREATE TABLE Telephone
(
    ref_firm INT NULL,
    telephone VARCHAR(50),
    ref_person INT NULL
);
```

```
ALTER TABLE Firm
ADD CONSTRAINT constr_Firm_Country
FOREIGN KEY (ref_country)
REFERENCES Country (id_country);
```

```
ALTER TABLE Telephone
ADD CONSTRAINT constr_Telephone_Firm
FOREIGN KEY (ref_firm)
REFERENCES Firm (id_firm);
```

```
ALTER TABLE Telephone
ADD CONSTRAINT constr_Telephone_Person
FOREIGN KEY (ref_person)
REFERENCES Person (id_person);
```

7. Вывести список фирм, их сотрудников, телефонов и стран проживания.

```
SELECT name_firm, Name, telephone, name_country FROM Telephone
INNER JOIN Firm ON ref_firm = id_firm
INNER JOIN Person ON ref_person = id_person
LEFT JOIN Country ON Person.ref_country = id_country;
```

8. Вывести список фирм и их сотрудников, а также фирм не имеющих сотрудников.

```
SELECT name_firm, Name FROM Telephone  
RIGHT JOIN Firm ON ref_firm = id_firm  
LEFT JOIN Person ON ref_person = id_person;
```

9. Вывести список фирм и стран в которых они находятся, их сотрудников с указанием страны регистрации.

```
SELECT name_firm, FC.name_country AS "Firm country", Name,  
PC.name_country AS "Residence country"  
FROM Telephone  
INNER JOIN Firm ON ref_firm = id_firm  
INNER JOIN Country FC ON Firm.ref_country = FC.id_country  
INNER JOIN Person ON ref_person = id_person  
LEFT JOIN Country PC ON Person.ref_country = PC.id_country;
```

Перечень запросов (индивидуальное задание)

1. Вывести названия проектов и имена дизайнеров, работающих над ними + имена дизайнеров, не работающих ни над одним проектом.

```
SELECT projects.name AS "project name", designers.full_name AS  
"name"  
FROM project_designers  
INNER JOIN projects ON ref_project = projects.id  
RIGHT JOIN designers ON ref_designer = designers.card_id;
```

2. Вывести количество программистов, работающих над каждым из проектов (в том числе над проектами без программистов).

```
SELECT projects.name AS "project name", count(ref_programmer)  
AS "programmers count"  
FROM project_programmers  
RIGHT JOIN projects ON ref_project = projects.id  
GROUP BY projects.name;
```

3. Вывести названия всех проектов не имеющих менеджера.

```
SELECT projects.name FROM project_managers  
RIGHT JOIN projects ON ref_project = projects.id  
WHERE ref_project IS NULL;
```