Федеральное государственное автономное образовательное учреждение

высшего образования

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа «Киберфизические системы и управление»

**Отчет №4**

по дисциплине «Базы данных на английском языке»

|  |  |  |
| --- | --- | --- |
| Выполнил:  студент гр. 3530902/00201 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Позолотин О. В. |
|  | <*подпись*> |  |
|  |  |  |
|  |  |  |
| Руководитель:  Кандидат т.н. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Нестеров С. А. |
|  | <*подпись*> |  |

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2023 г.

Санкт-Петербург

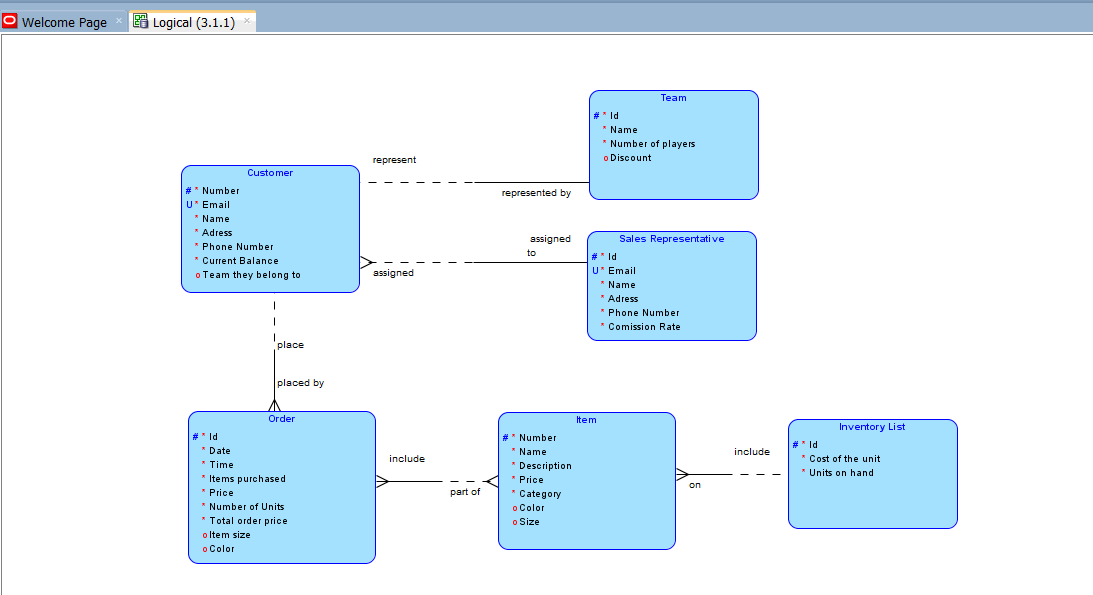
2023

**Ход работы**

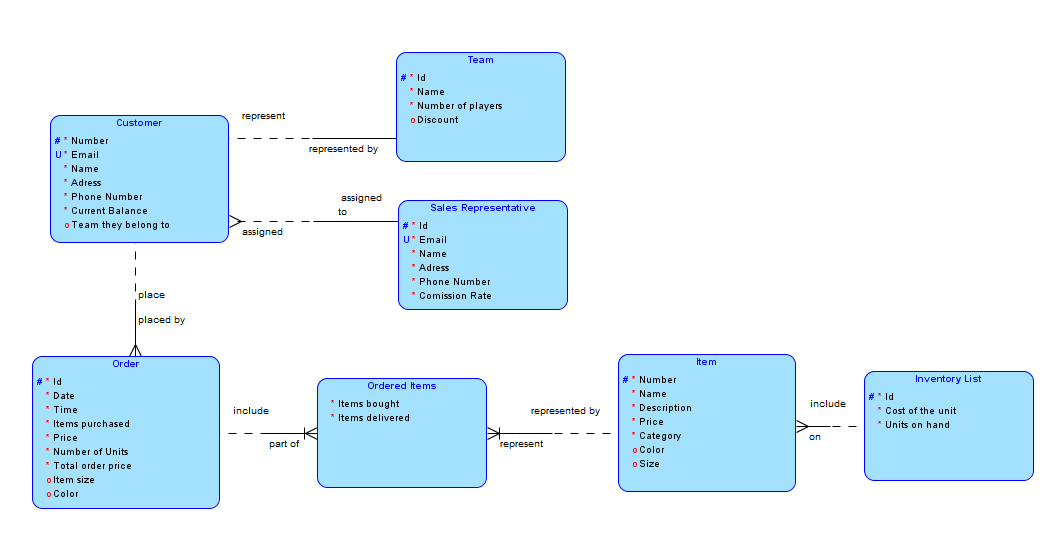
**Задание по презентации 3-1 More with Relationships**

**3\_1\_1**

Starting point we have:

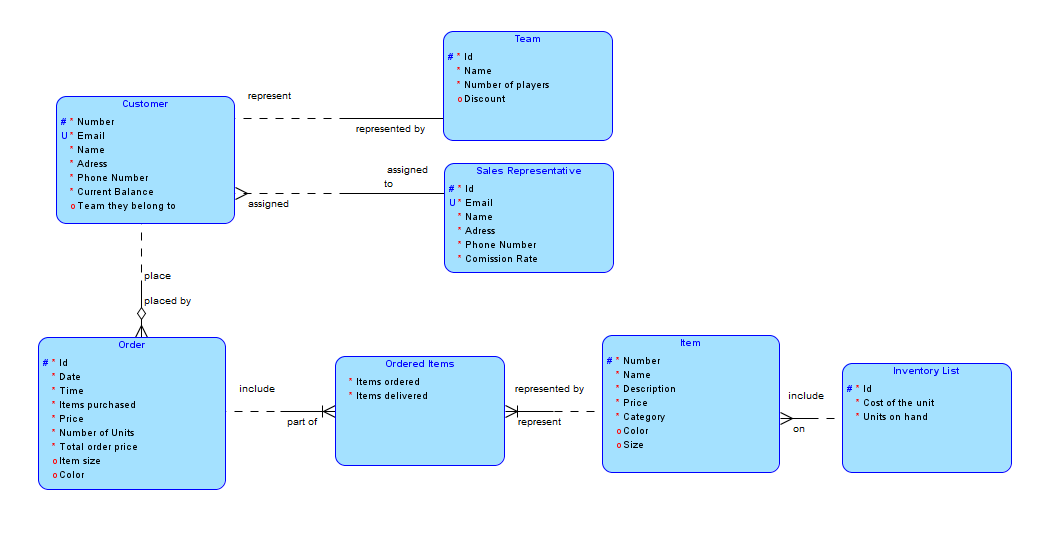


We have M:M relationship between Order and Item entities. Lets fix it:



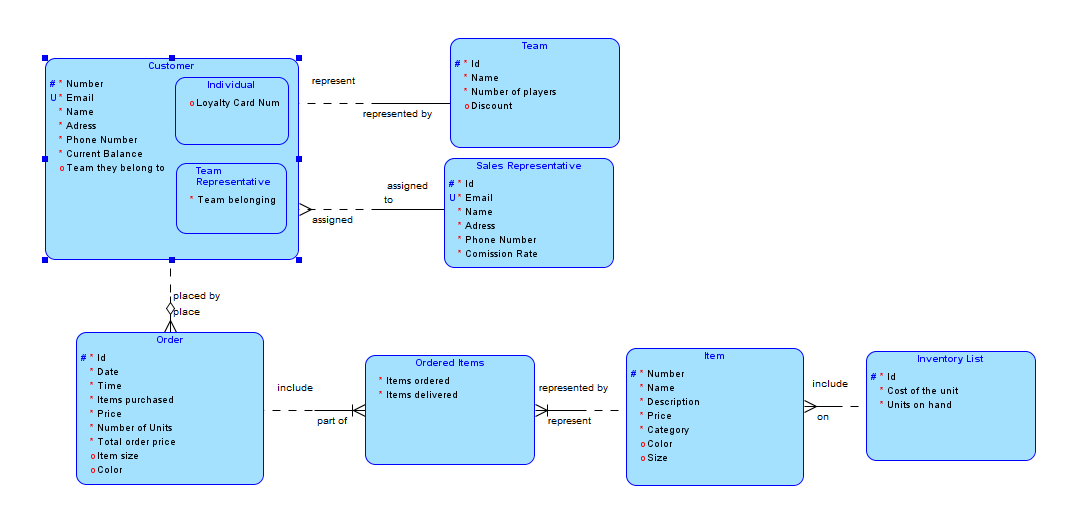
**Section 3 Lesson 1 Exercise 2: More with Relationships**

Identify and Illustrate Non-Transferable Relationships (S3L1 Objective 3)



**Section 3 Lesson 1 Exercise 3: More with Relationships**

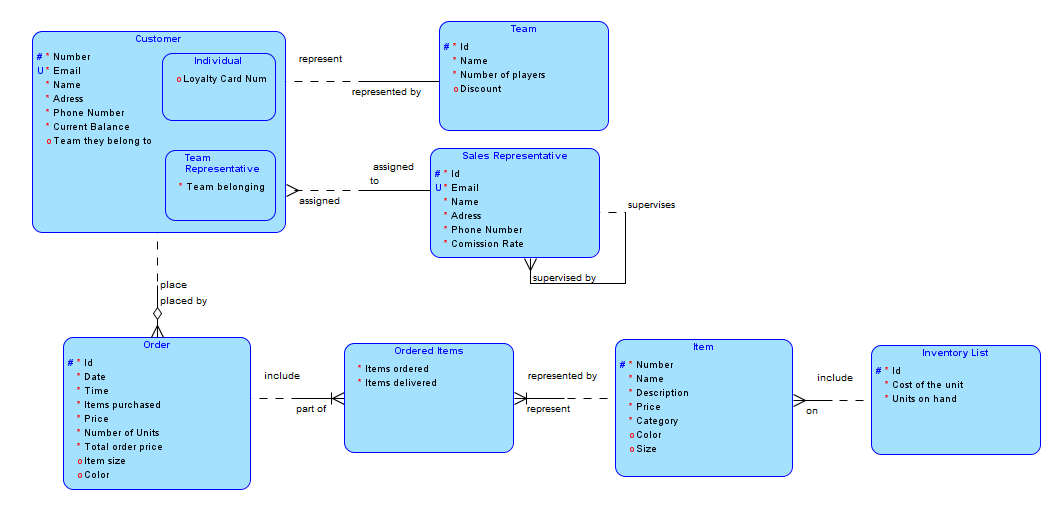
Identify and Draw Supertype and Subtype Entities



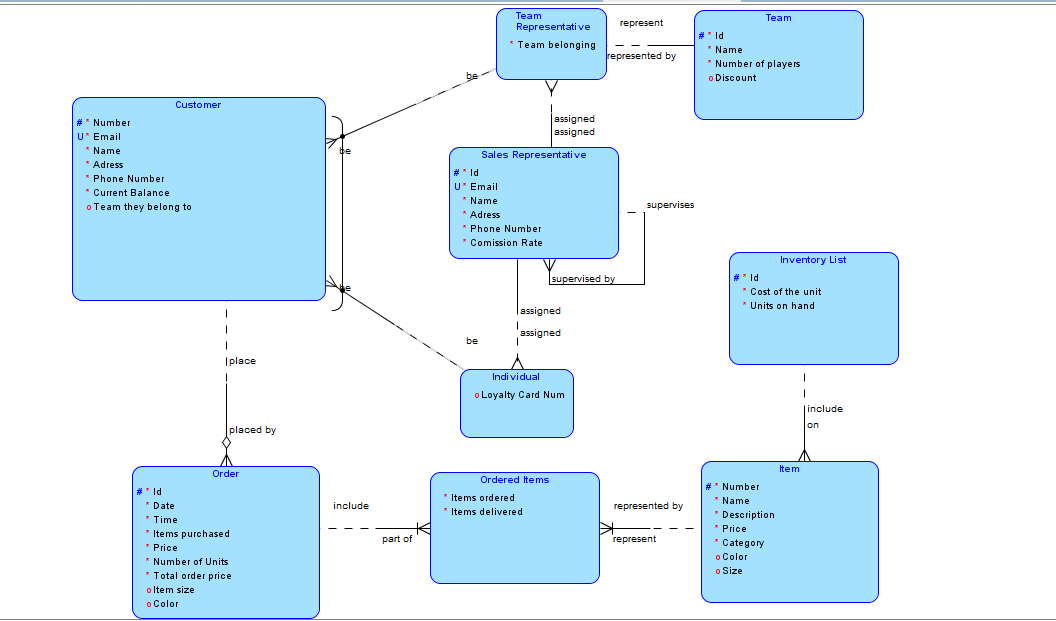
**Section 3 Lesson 1 Exercise 4: More with Relationships**

Identify Hierarchical, Recursive and Arc Relationships (S3L1 Objective 5)

**Part 1 - Recursive relationships**

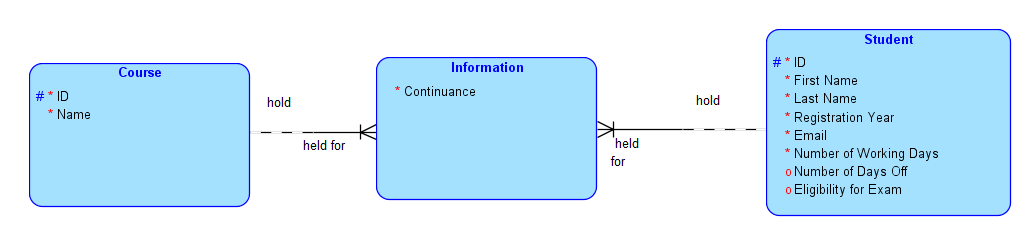
****

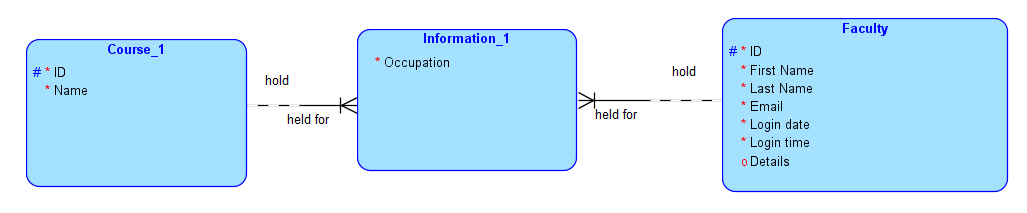
**Part 2 - Arc relationships**

****

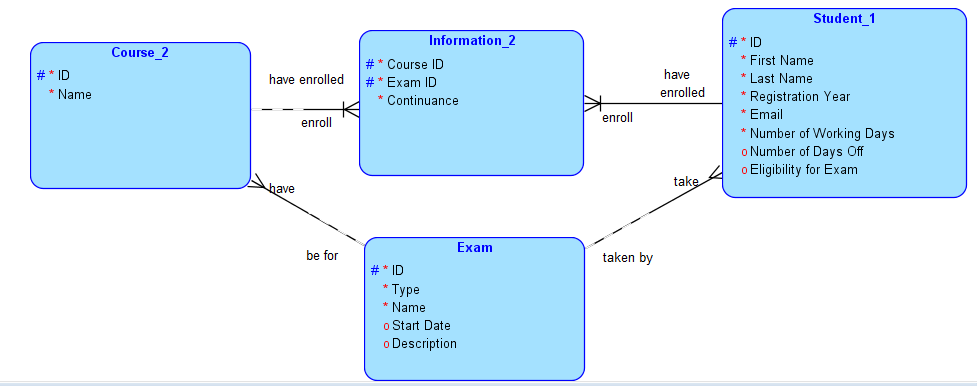
**3-1 More with Relationships**

**Resolve M: M relationships between STUDENT and the COURSE using a barred relationship.**

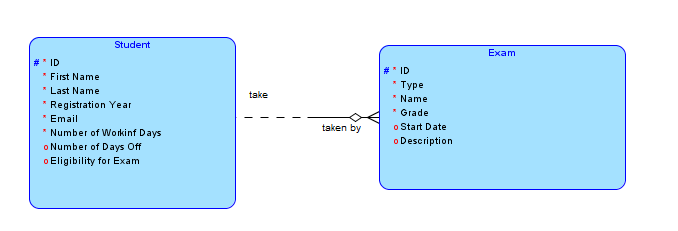
**Resolve M: M relationships between FACULTY and the COURSE.**



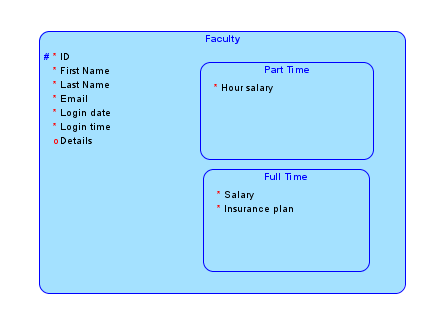
**Resolve M: M relationships between STUDENT,COURSE and EXAM.**

****

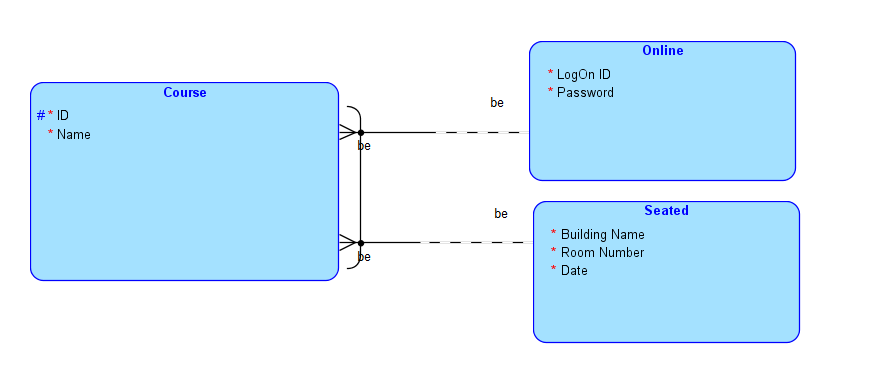
**Exercise 2: Adding nontransferability option to an ERD**

****

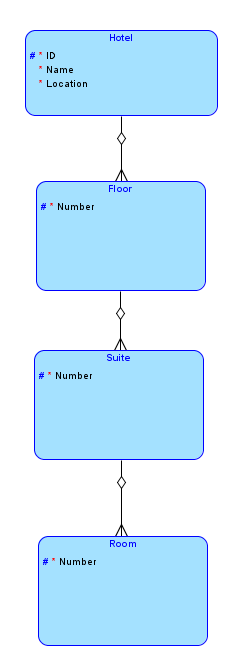
**Exercise 3: Identify and draw supertype and subtype entities**

****

**Exercise 4: Examine Exclusive Relationships (Arcs)**

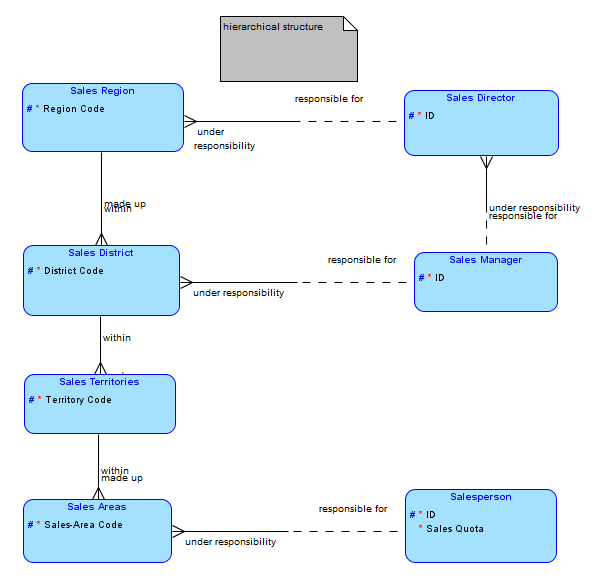
****

**Exercise 5: Model Hierarchical Data**

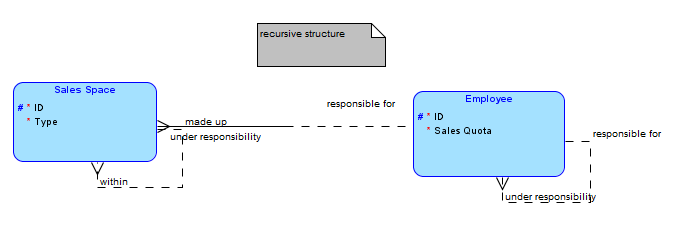
****

**Exercise 6: Model Hierarchical Data and Recursive Relationships**

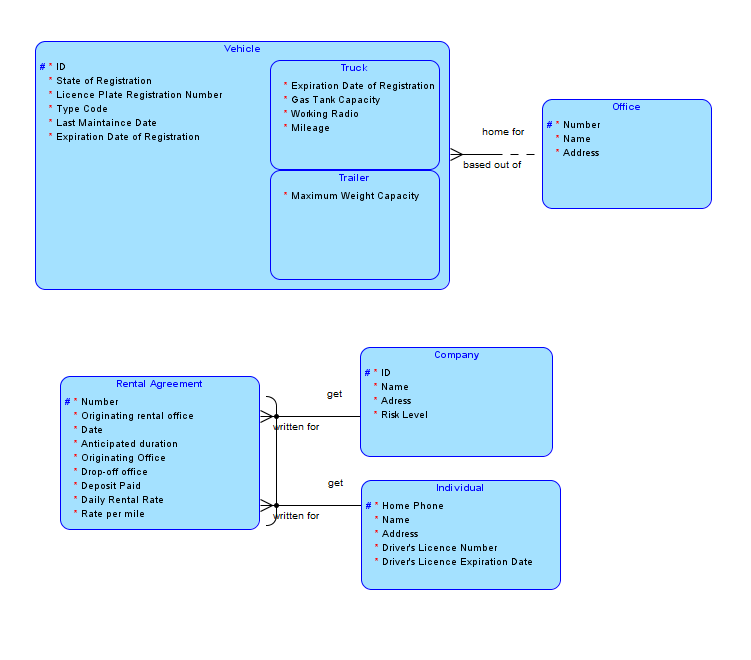
**Hierarchical structure**

****

**Recursive structure**

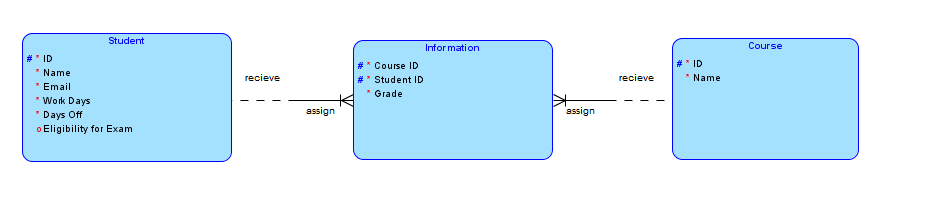
****

**Exercise 7: Developing a complete ERD using Supertype/Subtypes and Arcs**

****

**3-2 : Tracking Data Changes**

**Construct the ERD for the given scenario.**

****

**Examine the ERD that represents classroom assignments for different exams**

a. Why is start time part of the UID of ASSIGNMENT?

Because the start time determines which exam is being held.b. Name at least three time-related constraints. For example: End time must be later than start time. Indicate if the constraint represents conditional non-transferability.

b.Name at least three time-related constraints. For example: End time must be later than start time. Indicate if the constraint represents conditional non-transferability.

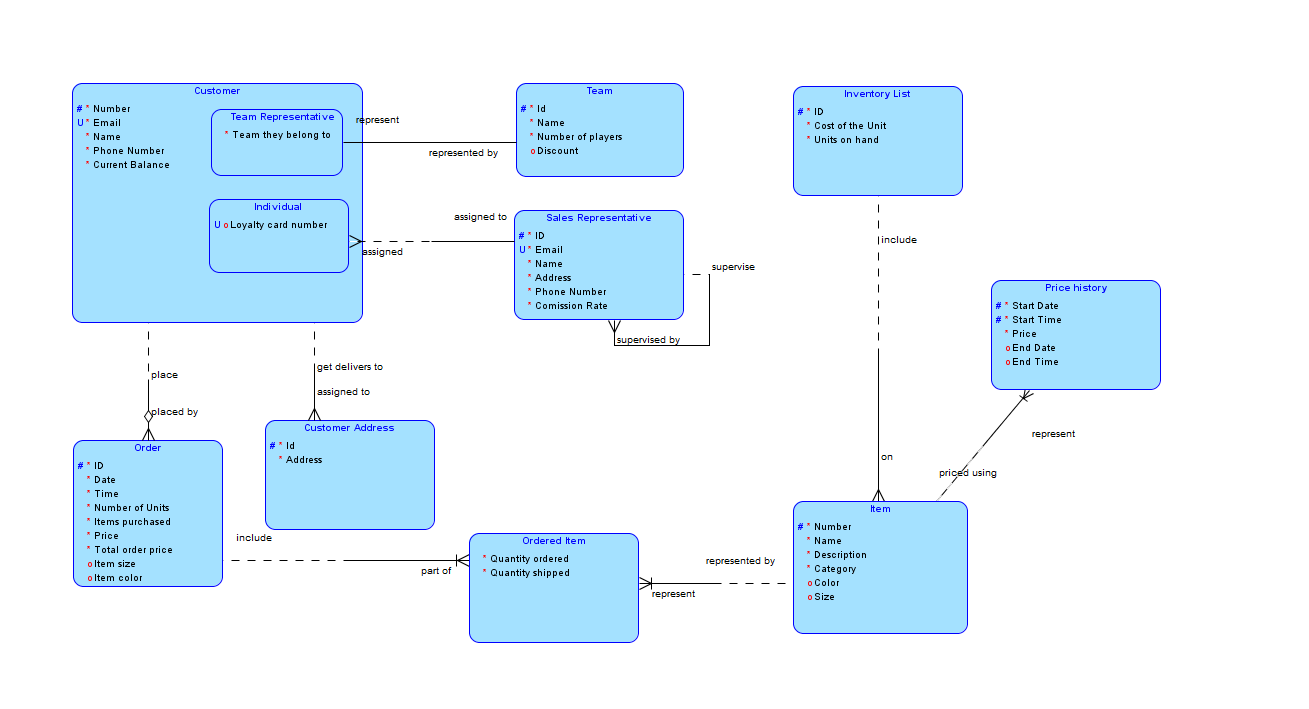
The start time cannot be less than the start time of the working day

The time of the end of the exam cannot be longer than the time of the end of the working day

It is impossible to conduct two exams simultaneously in one office

**Section 3 Lesson 2 Exercise : Tracking Data Changes**

**Keep Track of Data that Changes over Time (S3L2 Objective 1)**

****

**Лабораторный работы с курса «Управление данными»**

**Лабораторная работа 1**

**Customers**

Изображение выглядит как текст

Автоматически созданное описание

**Order Details**

**Изображение выглядит как стол

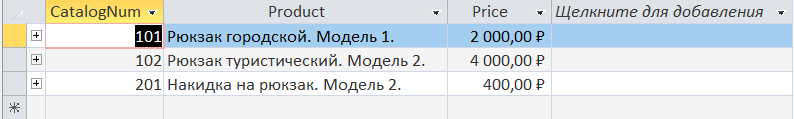
Автоматически созданное описание**

**Orders**

**Изображение выглядит как стол

Автоматически созданное описание**

**Products**

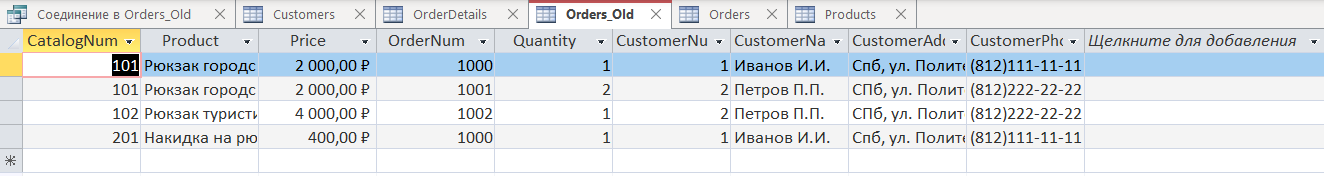
****

**Запрос**

**Изображение выглядит как стол

Автоматически созданное описание**

**Результат**

****

Имеется исходная таблица OrdersOld, не удовлетворяющая условиям 3НФ (поскольку, как минимум, она не удовлетворяет условиям 2НФ). Для того чтобы ее нормализовать, необходимо нормализовать к 2 НФ. Стоит отметить, что таблица приведена к 1НФ: в каждом атрибуте содержатся только скалярные значения и ни один атрибут не содержит NULL. Далее разделим таблицу на 4: Customer, OrderDetails, Order, Product и введем первичные ключи для каждой из них. Теперь атрибуты были неприводимо зависимы от первичного ключа. На этом этапе таблица удовлетворяет 2НФ. Поскольку неключевые атрибуты не зависят друг от друга и выполнены условия 2НФ, таблица приведена к 3НФ.

Запрос «Соединение в OrdersOld» демонстрирует, что соединение 4 сущностей дает нам исходную таблицу. Поскольку это выполняется, нормализация проведена успешно.

Если в исходную таблицу ввести составной ключ OrderNum и CatalogNum, то вторая нормальная форма выполняться не будет, потому что существует функциональная зависимость CatalogNum->Product

**Лабораторная работа 2**

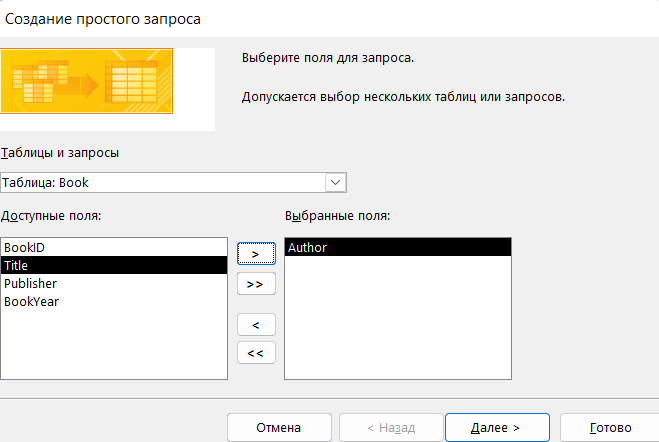
Задание. Используя базу lib.accdb в конструкторе создайте запрос, выводящий список авторов книг из таблицы Book (нам нужны только авторы, без названий книг и т.д.). Добавьте в таблицу данные таким образом, чтобы в ней было несколько книг одного автора. Выполните запрос и обратите внимание на то, что если автор написал несколько книжек, то в результатах запроса он будет фигурировать тоже несколько раз. Перейдите в режим SQL, в котором Вы увидите то предложение SQL, которое и является запросом: SELECT Book.Author FROM Book;

Добавим еще одну книгу Зеленина В.М.

**Изображение выглядит как текст

Автоматически созданное описание**

Создадим нужный нам запрос:

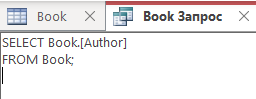


Выполним запрос:

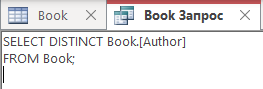
Изображение выглядит как стол

Автоматически созданное описание

Перейдем в режим SQL:



Преобразуем запрос так, чтоб не было повторяющихся авторов:

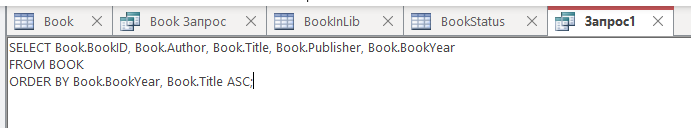


Выполнив, получим:

Изображение выглядит как текст, стол

Автоматически созданное описание

Задание. Напишите запрос, выводящий информацию о книгах из таблицы Book в порядке от самых старых к самым новым. Книги с одинаковым годом издания упорядочивайте в списке по названию в алфавитном порядке.



Выполним:

Изображение выглядит как текст

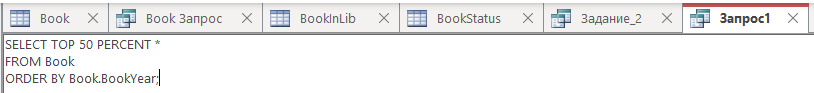
Автоматически созданное описание

Задание. Напишите запрос, выводящий информацию книгах, составляющих первую половину общего списка из таблицы Book, упорядоченного по возрастанию года (т.е. самые старые книги).

Добавим еще одну книгу для наглядности: Изображение выглядит как текст

Автоматически созданное описание

Напишем запрос:



Выполнив, получим:

Изображение выглядит как текст

Автоматически созданное описание

Задание. Скопируйте приведенный выше запрос в свою базу. Выполните его. При необходимости измените набор данных в таблицах так, чтобы в таблице Book были записи не связанные с записями из таблицы BookInLib. В запросе замените INNER JOIN на LEFT JOIN, проанализируйте результат. Объясните, почему для нашей структуры базы данных замена в этом запросе INNER JOIN на RIGHT JOIN не приведет к изменениям в выводимом наборе данных (тут надо вспомнить о внешних ключах, которые создавались в ходе выполнения первой лабораторной).

Выполним запрос из книги:

Изображение выглядит как стол

Автоматически созданное описание

LEFT JOIN

Изображение выглядит как стол

Автоматически созданное описание

RIGHT JOIN

Изображение выглядит как стол

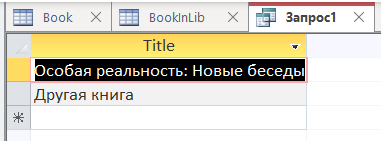
Автоматически созданное описание

Здесь все элементы BookInLib выведутся сразу. Замена не приведет к изменениям, поскольку в таблице не останется элементов.

Задание. Напишите на SQL запрос, который выведет в обратном алфавитном порядке названия книг, выпущенных после 1999 года, фамилия автора (значение поля Author) у которых начинается с буквы «Г» или название издательства заканчивается буквой «а».

Изображение выглядит как текст

Автоматически созданное описание



Задание. Напишите на SQL запрос, возвращающий список названий неиспользуемых статусов книг, то есть таких, что данный статус не назначен ни одной книге в таблице BookInLib. Подсказка: нужно использовать внешнее соединение таблиц и условие IS NULL. При необходимости отредактируйте используемый набор данных, чтобы такие статусы были, и проверьте работу запроса.

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как стол

Автоматически созданное описание

Задание. Опишите, что на ваш взгляд делает приведённый ниже запрос. Скопируйте его код в Access и выполните. Правильным ли оказалось ваше предположение?

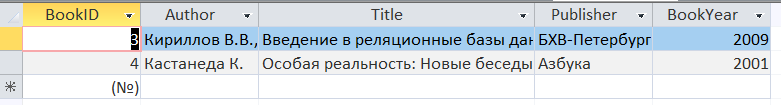
SELECT \*

FROM Book

WHERE Author LIKE 'А\*' OR BookYear> 2000 AND NOT

Publisher LIKE '[И,П]\*';

Такой запрос выведет информацию о книгах, у который либо имя автора начинается на буку ‘А’, либо книга издана после 2000 года, при всем при этом название издательства книги не начинается с букв ‘И’ или ‘П’.



Предположение оказалось верным.