**Занятие № 9**

**Номер учебной группы:** П-16

**Фамилия, инициалы учащегося:** Мозоль Павел Васильевич

**Дата выполнения работы:** 17.11.2022

**Тема работы:** «Разработка модели «сущность-связь» в нотации Чена с использованием современных CASE технологий»

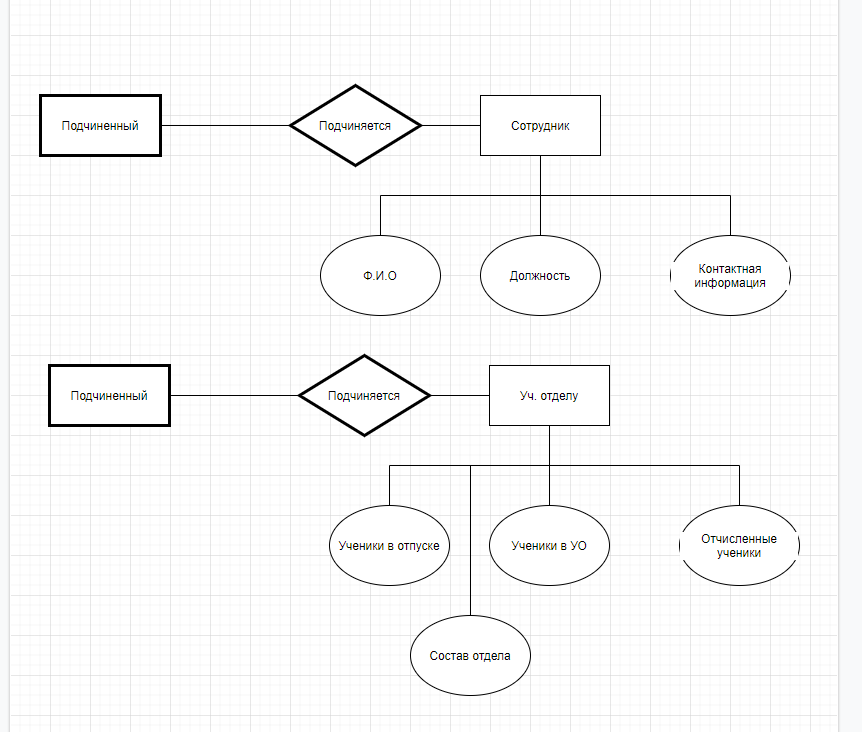
**Ход работы**

**Задание 1**

****

Связь соединяется с ассоциируемыми сущностями линиями. Возле каждой сущности на линии, соединяющей ее со связью, цифрами указывается класс принадлежности.

**Задание 2**

****

**Задание 3**

1. Логический и физический уровни модели данных в ERwin.

**Ответ: Логический уровень -** это абстрактный взгляд на данные, на нем данные представляются так, как выглядят в реальном мире, и могут называться так, как они называются в реальном мире, например, "Постоянный клиент", "Отдел" или "Фамилия сотрудника".

**Физическая модель** данных, напротив, зависит от конкретной СУБД, фактически являясь отображением системного каталога. В физической модели содержится информация о всех объектах БД.

1. Последовательность создания логической модели данных в ERwin.

**Ответ**: Различают три уровня логической модели, отличающихся по глубине представления информации о данных:

**А)** диаграмма сущность-связь (Entity Relationship Diagram, ERD);

**Б)** модель данных, основанная на ключах (Key Based model, KB);

**В)** полная атрибутивная модель (Fully Attributed model, FA).

1. Последовательность создания физической модели данных в ERwin.

**Ответ**: Различают два уровня физической модели:

* А) трансформационная модель (Transformation Model);
* Б) модель СУБД (DBMS Model).

1. Альтернативные ключи.

**Ответ**: Обработка входящих сообщений использует поля ключей объекта, чтобы найти существующую в системе запись для поддержки обновления и удаления. Обработка по умолчанию использует первичный ключ для извлечения существующих записей. Иногда первичный ключ неизвестен внешней системе, и задается альтернативный ключ (известный внешней системе) для поддержки обновлений и удалений записей внешней системой.

Обычно входящее сообщение использует первичный ключ объекта для поиска и обработки записей, уже существующих в системе. Однако иногда первичный ключ - это значение, сгенерированное внутренним образом, и оно недоступно внешней системе. В таких случаях можно задать альтернативный ключ для объекта; внешняя система передает поля альтернативных ключей, которые каркас интеграции использует для извлечения данных для этого объекта вместо полей первичных ключей.

Альтернативный ключ для объекта можно задать на уровне объекта или на уровне структуры объекта. Если альтернативный ключ задан на уровне объекта, то этот альтернативный ключ будут использовать все структуры объектов, содержащие объект. Если альтернативный ключ задан на уровне структуры объекта, то можно использовать разные структуры объектов, чтобы указать разные альтернативные ключи для каждого внешнего источника данных, используемого объектом. Индекс, составляющий альтернативный ключ для объекта, задается в программе Конфигурация базы данных. После создания индекса его можно выбрать в качестве альтернативного ключа для объекта.

1. Инвертированные индексы.

**Ответ: Инвертированный индекс** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *inverted index*) — [структура данных](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), в которой для каждого слова коллекции документов в соответствующем [списке](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) перечислены все документы в коллекции, в которых оно встретилось. Инвертированный индекс используется для поиска по текстам.

Есть два варианта инвертированного индекса:

* индекс, содержащий только список документов для каждого слова,
* индекс, дополнительно включающий позицию слова в каждом документ.

Опишем, как решается задача нахождения документов, в которых встречаются все слова из [поискового запроса](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B8%D1%81%D0%BA%D0%BE%D0%B2%D1%8B%D0%B9_%D0%B7%D0%B0%D0%BF%D1%80%D0%BE%D1%81). При обработке однословного поискового запроса ответ уже есть в инвертированном индексе — достаточно взять список, соответствующий слову из запроса. При обработке многословного запроса берётся пересечение списков, соответствующих каждому из слов запроса. Обычно в [поисковых системах](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B8%D1%81%D0%BA%D0%BE%D0%B2%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) после построения с помощью инвертированного индекса списка документов, содержащих слова из запроса, идет [ранжирование](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%BD%D0%B6%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_(%D0%BF%D0%BE%D0%B8%D1%81%D0%BA%D0%BE%D0%B2%D1%8B%D0%B5_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B)) документов из списка. Инвертированный индекс — это самая популярная структура данных, которая используется в [информационном поиске](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BF%D0%BE%D0%B8%D1%81%D0%BA).