**Практическое** **занятие** **№9**

**Группа**: П-15

**Учащийся**: Малохвей Дмитрий

**Дата**: 04.11.2021

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

**1. Цель работы:** Создание модели «сущность-связь» в нотации Чена с использованием современных CASE технологий

**2. Задание:** Разработать модели «сущность-связь» в нотации Чена с использованием современных CASE технологий

**Ход работы**

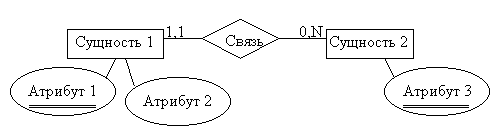
**Задание 1**

**Нотация Чена.**

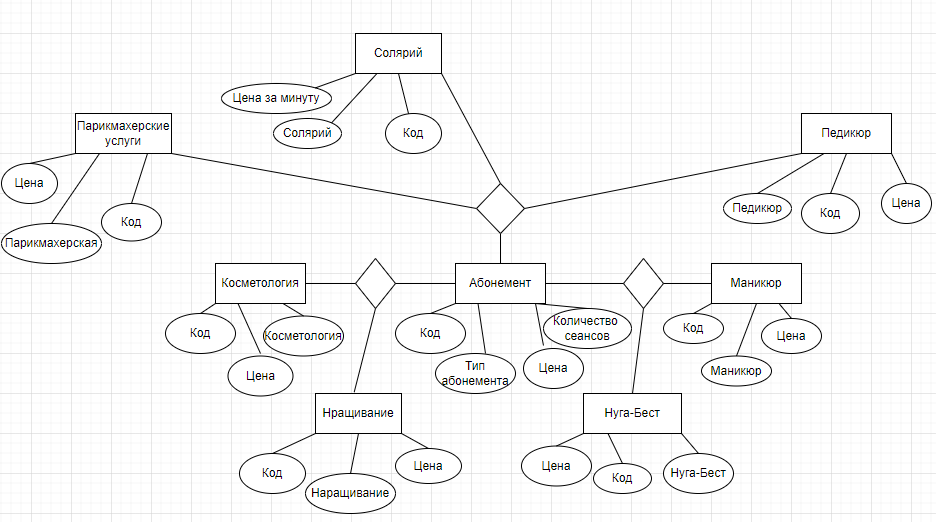
|  |  |
| --- | --- |
| **Элемент диаграммы** | **Обозначает** |
| http://www.mstu.edu.ru/study/materials/zelenkov/image144.gif | независимая сущность |
| http://www.mstu.edu.ru/study/materials/zelenkov/image41.gif | зависимая сущность |
| http://www.mstu.edu.ru/study/materials/zelenkov/image42.gif | родительская сущность в иерархической связи |
| http://www.mstu.edu.ru/study/materials/zelenkov/image145.gif | Связь |
| http://www.mstu.edu.ru/study/materials/zelenkov/image146.gif | идентифицирующая связь |
| http://www.mstu.edu.ru/study/materials/zelenkov/image123.gif | Атрибут |
| http://www.mstu.edu.ru/study/materials/zelenkov/image124.gif | первичный ключ |
| http://www.mstu.edu.ru/study/materials/zelenkov/image125.gif | внешний ключ (понятие внешнего ключа вводится в реляционной модели данных) |
| http://www.mstu.edu.ru/study/materials/zelenkov/image126.gif | многозначный атрибут |
| http://www.mstu.edu.ru/study/materials/zelenkov/image127.gif | получаемый (наследуемый) атрибут в иерархических связях |

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

**Пример:**



**Задание 2**

****

**Задание 3**

1. Логический и физический уровни модели данных в ERwin.

Ответ: ERwin имеет два уровня представления модели - логический и физический. 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. Поскольку стандартов на объекты БД не существует (например, нет стандарта на типы данных), физическая модель зависит от конкретной реализации СУБД. Следовательно, одной и той же логической модели могут соответствовать несколько разных физических моделей.

2. Последовательность создания логической модели данных в ERwin.

Ответ: Создание модели данных, как правило, начинается с создания логической модели. После описания логической модели, проектировщик может выбрать необходимую СУБД и ERwin автоматически создаст соответствующую физическую модель. На основе физической модели ERwin может сгенерировать системный каталог СУБД или соответствующий SQL-скрипт. Этот процесс называется прямым проектированием (Forward Engineering). Тем самым достигается масштабируемость - создав одну логическую модель данных, можно сгенерировать физические модели под любую поддерживаемую ERwin СУБД. С другой стороны, ERwin способен по содержимому системного каталога или SQL-скрипту воссоздать физическую и логическую модель данных (Reverse Engineering).

3. Последовательность создания физической модели данных в ERwin.

Ответ: На основе полученной логической модели данных можно сгенерировать физическую модель для другой СУБД и затем сгенерировать ее системный каталог. Следовательно, ERwin позволяет решить задачу по переносу структуры данных с одного сервера на другой. Например, можно перенести структуру данных с Oracle на Informix (или наоборот) или перенести структуру dbf-файлов в реляционную СУБД, тем самым облегчив решение по переходу от файл-серверной к клиент-серверной ИС. Заметим, однако, что формальный перенос структуры "плоских" таблиц на реляционную СУБД обычно неэффективен. Для того чтобы извлечь выгоды от перехода на клиент-серверную технологию, структуру данных следует модифицировать.

4. Альтернативные ключи.

Ответ: Альтернативный ключ - это атрибут (или группа атрибутов), несовпадающий с первичным ключом и уникально идентифицирующий экземпляр сущности. Например, для сущности Пользователь (Код пользователя, фамилия. имя, отчество) группа атрибутов “фамилия”, “имя”, “отчество” может являться альтернативным ключом (в предположении, что на предприятии не работают полные тезки).

5. Инвертированные индексы.

Ответ: Инвертированный индекс (англ. inverted index) — структура данных, в которой для каждого слова коллекции документов в соответствующем списке перечислены все документы в коллекции, в которых оно встретилось. Инвертированный индекс используется для поиска по текстам.