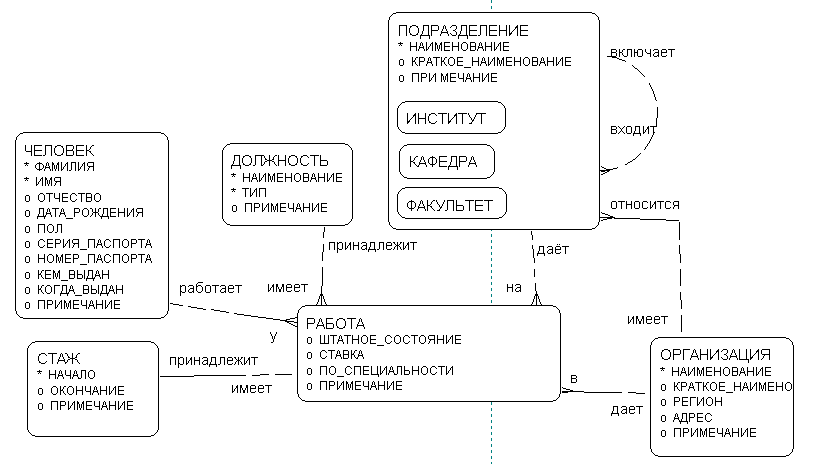
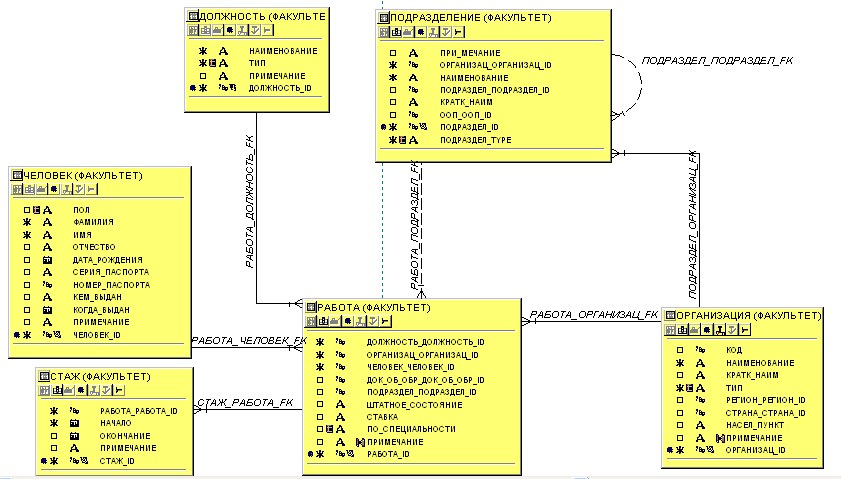
## Методы представления специализаций в реляционной схеме Oracle Designer

Oracle Designer представляет собой интегрированную CASE-среду для автоматизации процессов всех этапов жизненного цикла сложной прикладной системы. Так же Oracle обеспечивает гибкость при проектировании схемы ПрО за счёт предоставления инструментальных средств поддержки модели типа "сущность-связь" (ER) в нотации Баркера (ERB). Отталкиваясь от созданной ERВ-схемы преобразователь проекта БД генерирует реляционную схему, автоматизируя, тем самым, процесс преобразования ERВ-схемы в реляционную. Сами правила трансформации реализованы в Oracle Designer в компоненте DDT. Стивен Бобровски [1] описывает представление реляционной схемы в Oracle, а сам процесс генерации реляционной схемы из ERВ-схемы описан в книге Lilian Hobbs и Susan Hillson [2**.**]. На рисунке 1 приведён пример представления в виде ERВ-схемы Oracle Designer связи типа «работа». На рисунке 2 представлена соответствующая реляционная схема, полученная DDT автоматически. Поскольку объектом исследования данной работы являются специализации, то в данном разделе рассмотрим, какие методы используются для преобразования этих конструкций из ERВ-схемы в реляционную схему в Oracle Designer [3].



**Рисунок 1. ERB-схема. Связь «работа»**



**Рисунок 2. Реляционная схема. Связь «работа»**

1 метод: *Single table implementation* (одно отношение)

В этом методе создается одно отношение для представления и множества сущностей суперкласса и всех множеств сущностей его подклассов.

1. Отношения: Создаётся одно отношение, независимо от количества подклассов.
2. Атрибуты: Созданное отношение включает однозначные атрибуты множества сущностей суперкласса и всех множеств сущностей его подклассов, а также вводится дополнительный атрибут «тип подкласса».
3. Ограничения целостности:
   1. Унаследованные от семантической схемы
   * NULL / NOT NULL

Унаследованы от атрибутов множества сущностей суперкласса.

* + PRIMARY KEY и UNIQUE

PRIMARY KEY унаследовано от множества сущностей суперкласса.

UNIQUE унаследовано от множества сущностей суперкласса и подклассов.

* + CHECK

Унаследовано от множества сущностей суперкласса и его подклассов.

* 1. Порожденные методом реализации.
  + NULL / NOT NULL

Все атрибуты, образованные от множества сущностей подклассов и внешние ключи, образованные от множества связей множества сущностей подклассов в созданном отношении имеют ограничение NULL.

Дополнительный атрибут «тип подкласса» в созданном отношении имеет ограничение NOT NULL.

* + PRIMARY KEY и UNIQUE

PRIMARY KEY не создаются, т.к. унаследованы от атрибутов множества сущностей суперкласса.

Создаётся ограничение UNIQUE для первичных ключей множества сущностей подклассов, которые в ERB-схеме имели ограничение PRIMARY KEY.

* + FOREIGN KEY

Создаются внешние ключи, образованные от множества связей множества сущностей суперкласса, с исходным ограничением NULL / NOT NULL и от множества связей множества сущностей подклассов.

* + CHECK

Дополнительная проверка CHECK на ограничение NOT NULL для атрибутов множества сущностей подкласса, которые в ERB-схеме имели данное ограничение.

На рисунке 3, представлена специализация, которая делит множество работников (EMPLOYEE) на сотрудников, работающих полный (FULL TIME) и неполный рабочий день (PART TIME). Атрибут *зарплата* (salary) является обязательным для работников, занятых полный рабочий день, а атрибут *почасовая ставка* (hourly rate) является обязательным для работников, занятых неполный рабочий день. При преобразовании данным методом в реляционную модель, эти атрибуты становятся необязательными.

Для данного примера, код ограничения CHECK будет выглядеть следующим образом:

CHECK (epe\_type = ‘FTE’ and salary is not null and hourly\_rate is null and agy\_id is null) OR (epe\_type = ‘PTE’ and salary is null and hourly\_rate is not null and agy\_id is not null), где:

epe\_type – тип сотрудника, который может работать полный рабочий день (FTE) или неполный рабочий день (PTE);

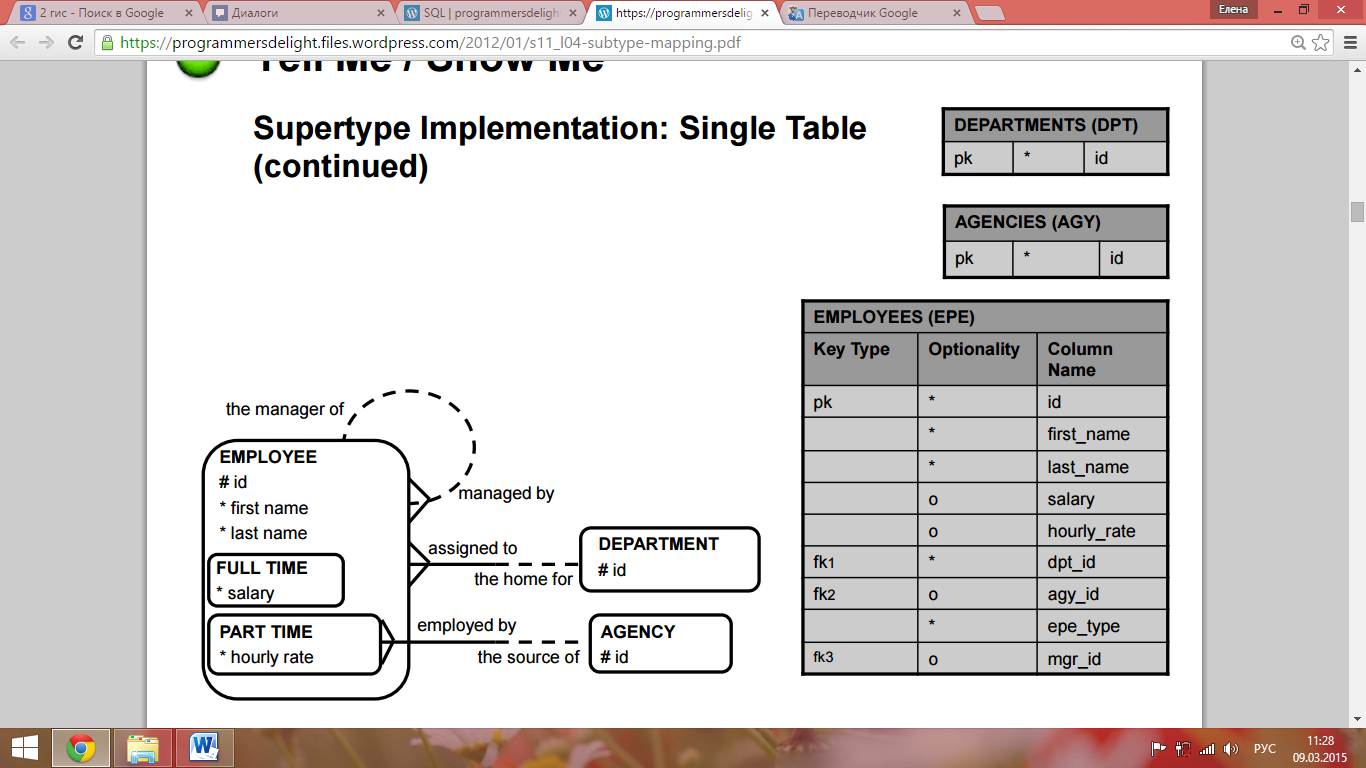
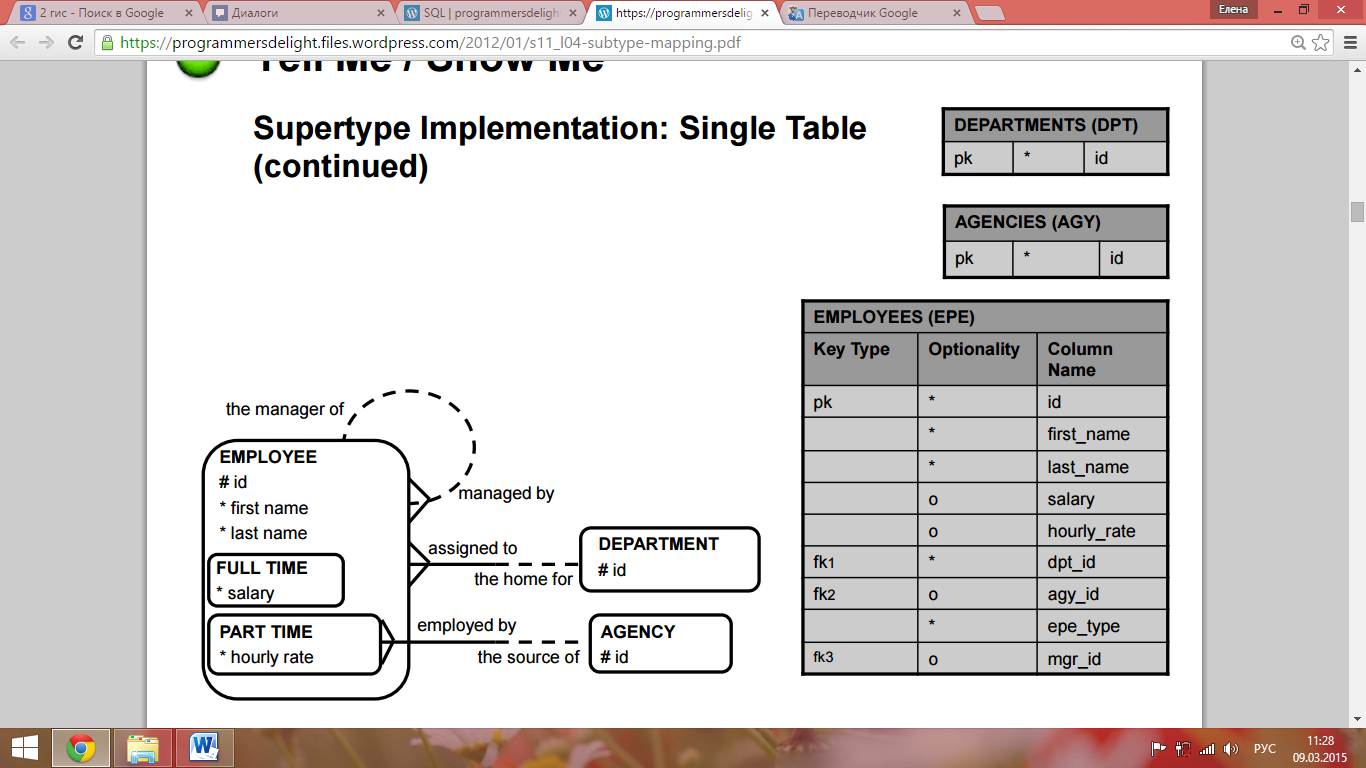
salary – зарплата сотрудника, работающего полный рабочий день;

hourly\_rate – почасовая ставка сотрудника, работающего неполный рабочий день;

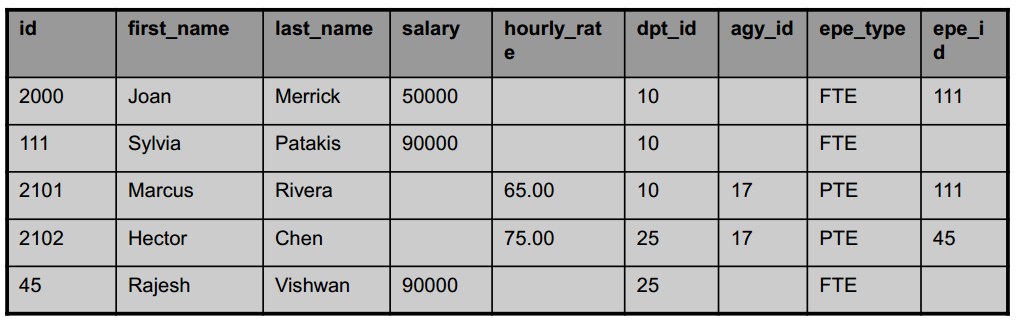
agy\_id – идентификатор агентства, которое направило сотрудника, работающего неполный рабочий день.

Код проверяет, что сотрудник, который работает полный рабочий день (epe\_type = 'FTE') должен иметь ненулевое значение атрибута salary, а атрибуты hourly\_rate и agy\_id должны быть пустыми. И наоборот, сотрудник, который работает неполный рабочий день (epe\_type = 'PTE') должен иметь ненулевое значение атрибутов hourly\_rate и agy\_id, но атрибут salary должен быть оставлен пустым.

Вид таблицы, созданной по данному методу и пример её заполнения представлен на рисунке 4.



**Рисунок 3. Метод 1 «Single table implementation». Специализация сотрудников**



**Рисунок 4. Метод 1 «Single table implementation». Пример данных для специализации сотрудников.**

Данный метод необходимо использовать, когда:

* + множества сущностей суперкласса имеют больше атрибутов, чем множества сущностей подклассов;
  + большинство связей на уровне суперкласса;
  + большинство сущностей ПрО принадлежат более чем одному подклассу одновременно.

Именно этим методом в примере на рисунке 2 была преобразована специализация, которая делит множество *подразделений* на *институты, кафедры и факультеты*.

2 метод: *Separate table implementation* (отношения только для подклассов)

В этом методе создаются только отношения для всех подклассов, а атрибуты множества сущностей суперкласса добавляются в каждое построенное отношение.

1. Отношения: Создаются отношения для подклассов.
2. Атрибуты: Каждое созданное отношение включает однозначные атрибуты множества сущностей суперкласса и соответствующего подкласса.
3. Ограничения целостности:
   1. Унаследованные от семантической схемы
   * NULL / NOT NULL

Унаследованы от атрибутов множества сущностей суперкласса и соответствующего подкласса.

* + PRIMARY KEY и UNIQUE

PRIMARY KEY унаследовано от множества сущностей соответствующего подкласса.

UNIQUE унаследовано от множества сущностей суперкласса и подклассов.

* + CHECK

Унаследовано от множества сущностей суперкласса и соответствующего подкласса.

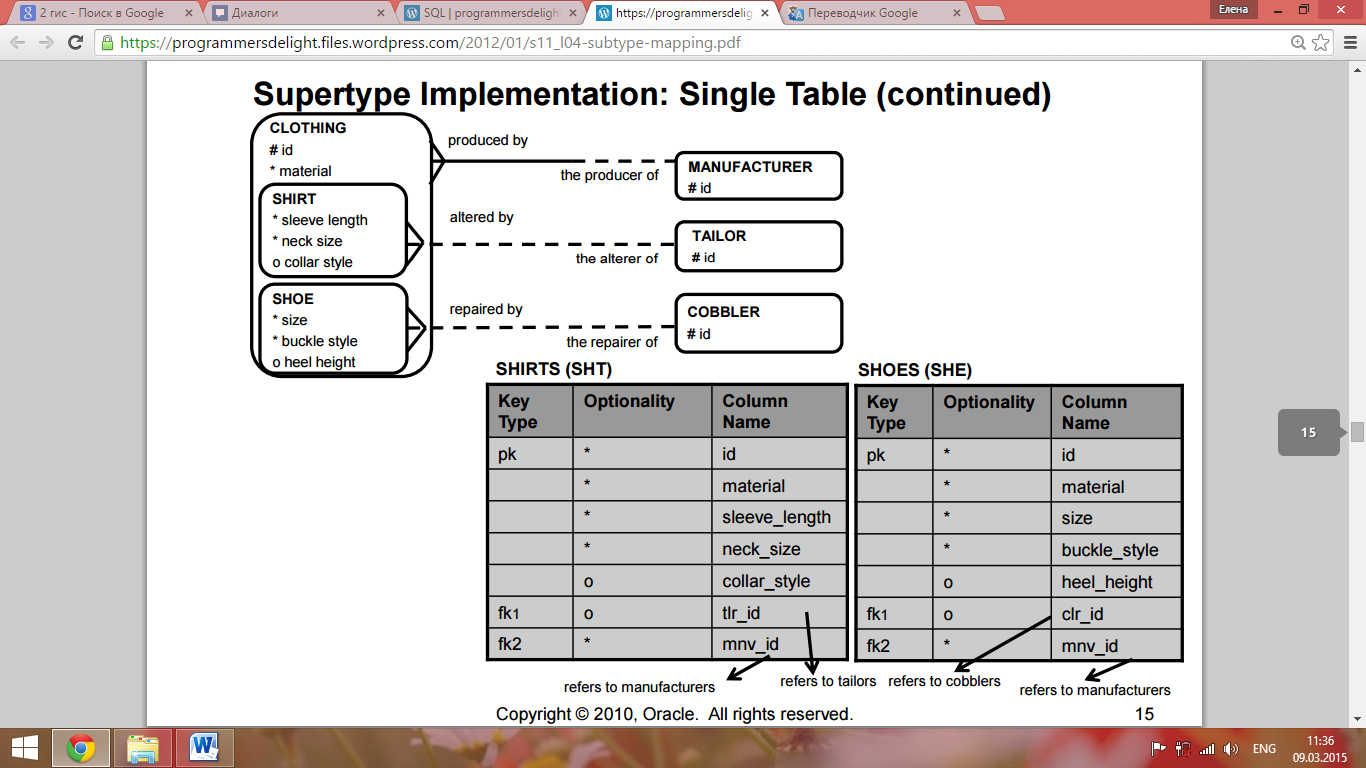
* 1. Порожденные методом реализации
  + Ограничений NULL / NOT NULL, PRIMARY KEY и CHECK порождённых методом реализации, - нет, поскольку они полностью унаследованы от множества сущностей суперкласса или подклассов.
  + UNIQUE

Если ограничение PRIMARY KEY для множества сущностей суперкласса в ERB-схеме было естественным первичным ключом, то в каждом созданном отношении это ограничение PRIMARY KEY заменяется на ограничение UNIQUE.

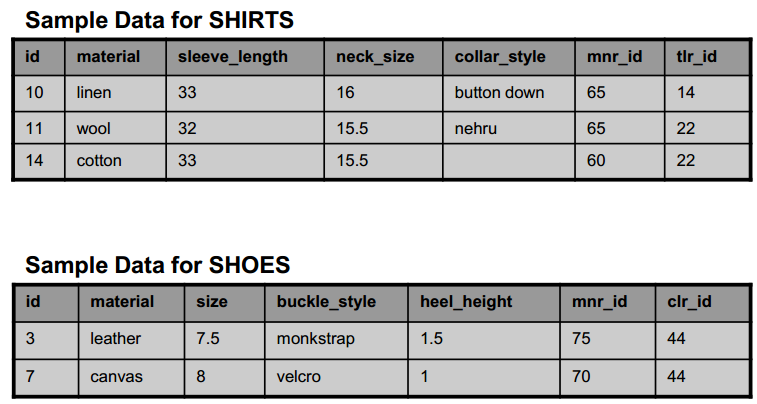
* + FOREIGN KEY

В каждое созданное отношение добавляются внешние ключи, образованные от связей множества сущностей суперкласса и соответствующего подкласса с исходным ограничением NULL / NOT NULL.

На рисунке 5 представлена специализация, которая выделяет множество рубашек (SHIRT) и обуви (SHOE) из множества одежды (CLOTHING). Вид таблиц, созданных по данному методу и пример их заполнения представлен на рисунке 6.



**Рисунок 5. Метод 2 «Separate table implementation». Специализация одежды**



**Рисунок 6. Метод 2 «Separate table implementation». Пример данных для специализации одежды**

Данный метод необходимо использовать, когда:

* + множества сущностей суперкласса имеют меньше атрибутов, чем множества сущностей подклассов;
  + большинство связей на уровне подклассов;
  + в ПрО не существует сущностей, принадлежащих только множеству сущностей суперкласса;
  + большинство сущностей ПрО являются членами только одного из подклассов.

3 метод: *Arc implementation* (отношение для всех множеств сущностей; внешние ключи в суперклассе)

В этом методе создается одно отношение для множества сущностей суперкласса и по одному отношению для каждого множества сущностей подклассов. Суперкласс связан с подклассами с помощью внешних ключей, которые определятся в отношении суперкласса.

1. Отношения: Создается одно отношение для множества сущностей суперкласса и по одному отношению для каждого множества сущностей подклассов.
2. Атрибуты: Каждое созданное отношение включает только однозначные атрибуты множества сущностей, на основе которого оно создано.
3. Ограничения целостности:
   1. Унаследованные от семантической схемы
   * NULL / NOT NULL

Унаследованы от атрибутов множества сущностей суперкласса или подклассов.

* + PRIMARY KEY и UNIQUE

Унаследованы от множества сущностей суперкласса или подклассов.

* + CHECK

Унаследовано от соответствующего множества сущностей.

* 1. Порожденные методом реализации
  + Ограничений NULL / NOT NULL, PRIMARY KEY и CHECK, порождённых методом реализации, - нет, поскольку они полностью унаследованы от множества сущностей суперкласса или подклассов.
  + FOREIGN KEY

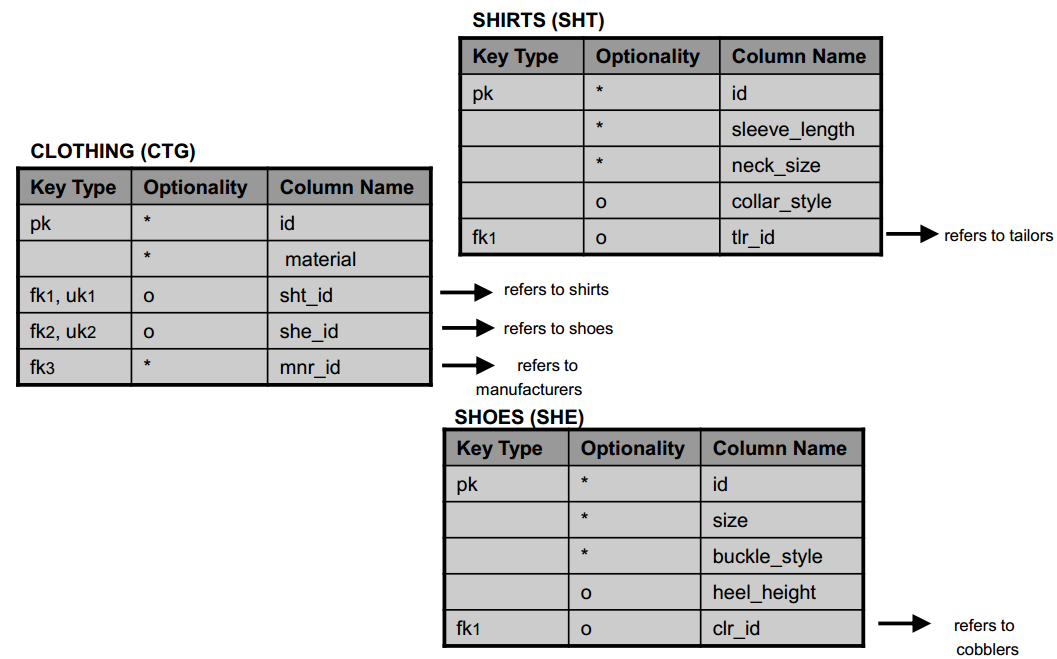
В отношения, построенные для множества сущностей подклассов, добавляются внешние ключи, образованные от связей множества сущностей подклассов с исходным ограничением NULL / NOT NULL.

В отношении, построенном для множества сущностей суперкласса, добавляются внешние ключи, образованные от связей множества сущностей суперкласса, а также создаются внешние ключи для связи с отношениями, построенными для множества сущностей подклассов.

* + UNIQUE

Для внешних ключей, созданных для связи с множеством сущностей подклассов в отношении, построенном для множества сущностей суперкласса, добавляются ограничения UNIQUE с ограничением NULL.

На рисунке 7 представлен тот же пример специализации, что и на рисунке 5. Специализация выделяет множество рубашек (SHIRT) и обуви (SHOE) из множества одежды (CLOTHING). Вид таблиц, созданных данным методом и пример их заполнения представлен на рисунке 8.



**Рисунок 7. Метод 3 «Arc implementation». Специализация одежды**

**Sample Data for CLOTHING**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id** | **material** | **sht\_id** | **she\_id** | **mnr\_id** |
| **3** | leather |  | 1 | 75 |
| **7** | canvas |  | 2 |  |
| **10** | linen | 1 |  | 65 |
| **11** | wool | 2 |  | 65 |
| **14** | cotton | 3 |  | 60 |

**Sample Data for SHIRTS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id** | **sleeve\_length** | **neck\_size** | **collar\_style** | **tlr\_id** |
| **1** | 33 | 16 | button down | 14 |
| **2** | 32 | 15.5 | nehru | 22 |
| **3** | 33 | 15.5 |  | 22 |

**Sample Data for SHOES**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id** | **size** | **buckle\_style** | **heel\_height** | **clr\_id** |
| **1** | 7.5 | monkstrap | 1.5 | 44 |
| **2** | 8 | velcro | 1 | 44 |

**Рисунок 8. Метод 3 «Arc implementation». Пример данных для специализации одежды**

Данный метод необходимо использовать, когда:

* + множества сущностей каждого подкласса представляют информацию, которая может быть использована самостоятельно, без информации, представленной в суперклассе;
  + множества сущностей суперкласса и подклассов имеют много атрибутов и связей.

4 метод: *Implicit sub-type implementation* (отношения для всех множеств сущностей; в отношениях подклассов дублируются атрибуты суперкласса)

В этом методе создается одно отношение для множества сущностей суперкласса и по одному отношению для каждого множества сущностей подклассов с дублированием атрибутов множества сущностей суперкласса в подклассах.

1. Отношения: Создается одно отношение для множества сущностей суперкласса и по одному отношению для каждого множества сущностей подклассов.
2. Атрибуты: Отношения, созданные для множества сущностей подклассов, включают свои атрибуты, а также однозначные атрибуты множеств сущностей суперкласса. Отношение, созданное для множества сущностей суперкласса, включает только свои атрибуты.
3. Ограничения целостности:
   1. Унаследованные от семантической схемы
   * NULL / NOT NULL

Для отношения, построенного для множества сущностей суперкласса, ограничения NULL / NOT NULL унаследованы от атрибутов множества сущностей суперкласса.

Для отношений, построенных для множества сущностей подклассов, ограничения NULL / NOT NULL унаследованы от атрибутов множества сущностей суперкласса и соответствующего подкласса.

* + PRIMARY KEY и UNIQUE

Для отношения, построенного для множества сущностей суперкласса, ограничения PRIMARY KEY и UNIQUE унаследованы от множества сущностей суперкласса.

Для отношений, построенных для множества сущностей подклассов, ограничение PRIMARY KEY унаследовано от множества сущностей соответствующего подкласса, а ограничение UNIQUE унаследовано и от множества сущностей суперкласса и от соответствующего подкласса.

* + CHECK

Для отношения, построенного для множества сущностей суперкласса, ограничение CHECK унаследовано от множества сущностей суперкласса.

Для отношений, построенных для множества сущностей подклассов, ограничение CHECK унаследовано от множества сущностей суперкласса и соответствующего подкласса.

* 1. Порожденные методом реализации
  + Ограничений NULL / NOT NULL, PRIMARY KEY и CHECK, порождённых методом реализации, - нет, поскольку они полностью унаследованы от множества сущностей суперкласса или суперкласса и соответствующего подклассов.
  + UNIQUE

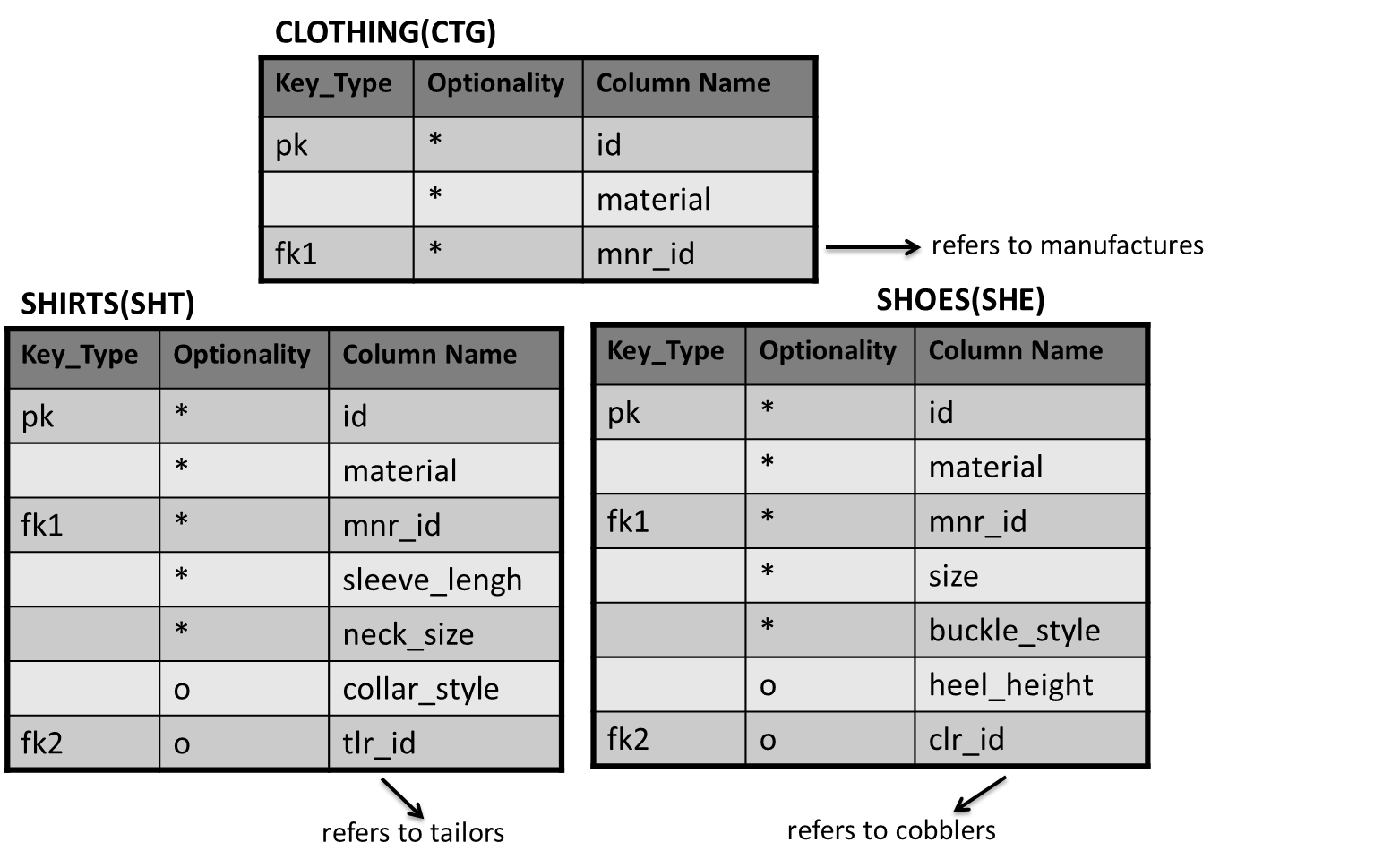
Если ограничение PRIMARY KEY для множества сущностей суперкласса в ERB-схеме было естественным первичным ключом, то в каждом отношении, созданном для множества сущностей подклассов это ограничение дублируется в отношение и заменяется на ограничение UNIQUE.

* + FOREIGN KEY

В отношение, построенное для множества сущностей суперкласса, добавляются внешние ключи, образованные от связей множества сущностей суперкласса, с исходным ограничением NULL / NOT NULL.

В отношения, построенные для множества сущностей подклассов, добавляются внешние ключи, образованные от связей множества сущностей суперкласса и соответствующего подкласса, с исходным ограничением NULL / NOT NULL.

На рисунке 9 представлен тот же пример специализации, что и на рисунке 5. Специализация выделяет множество рубашек (SHIRT) и обуви (SHOE) из множества одежды (CLOTHING). Вид таблиц, созданных по данному методу и пример их заполнения представлен на рисунке10.



**Рисунок 9. Метод 4 «Implicit sub-type implementation». Специализация одежды**

**Sample Data for CLOTHING**

|  |  |  |
| --- | --- | --- |
| **Id** | **material** | **mnr\_id** |
| **1** | leather | 35 |
| **2** | canvas |  |
| **3** | cotton | 22 |
| **4** | wool | 35 |

**Sample Data for SHIRTS**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **id** | **material** | **mnr\_id** | **sleeve\_length** | **neck\_size** | **collar\_style** | **tlr\_id** |
| **1** | linen | 65 | 33 | 16 | button down | 14 |
| **2** | wool |  | 32 | 15.5 | nehru | 22 |
| **3** | cotton | 60 | 33 | 15.5 |  | 22 |

**Sample Data for SHOES**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **id** | **Material** | **mnr\_id** | **size** | **buckle\_style** | **heel\_height** | **clr\_id** |
| **1** | leather | 75 | 7.5 | monkstrap | 1.5 | 44 |
| **2** | canvas |  | 8 | velcro | 1 | 44 |

**Рисунок 10. Метод 4 «Implicit sub-type implementation». Пример данных для специализации одежды**

Данный метод необходимо использовать, когда:

* + множества сущностей суперкласса имеют существенно меньше атрибутов и связей, чем множества сущностей подклассов;
  + в ПрО существуют сущности, принадлежащие только множеству сущностей суперкласса;
  + большинство сущностей ПрО являются членами только одного из подклассов.

# Список использованной литературы

1. *Бобровски С.* Oracle 7 вычисления клиент/сервер .- пер. с англ. М.: Лори, 1996. - 651с.
2. *Hobbs* *L.* Oracle 8i warehousing / *L.Hobbs, S. Hillson*.-USA.: Butterworth-Heinemann, 2000. -199 с.
3. *Oracle Academy*.-2011. <https://programmersdelight.wordpress.com/category/sql/> URL:<https://programmersdelight.files.wordpress.com/2012/01/s11_l04-subtype-mapping.pdf> (дата обращения 15.11.2014)<https://programmersdelight.wordpress.com/category/sql/><https://programmersdelight.files.wordpress.com/2012/01/s11_l04-subtype-mapping.pdf>https://programmersdelight.files.wordpress.com/2012/01/s11\_l04-subtype-mapping.pdf