



Факультет программной инженерии и компьютерной техники
Информационные системы и базы данных

Лабораторная работа № 2

Вариант 1134

Преподаватель: Харитонов А. Е.

Выполнила: Блинова М.А.

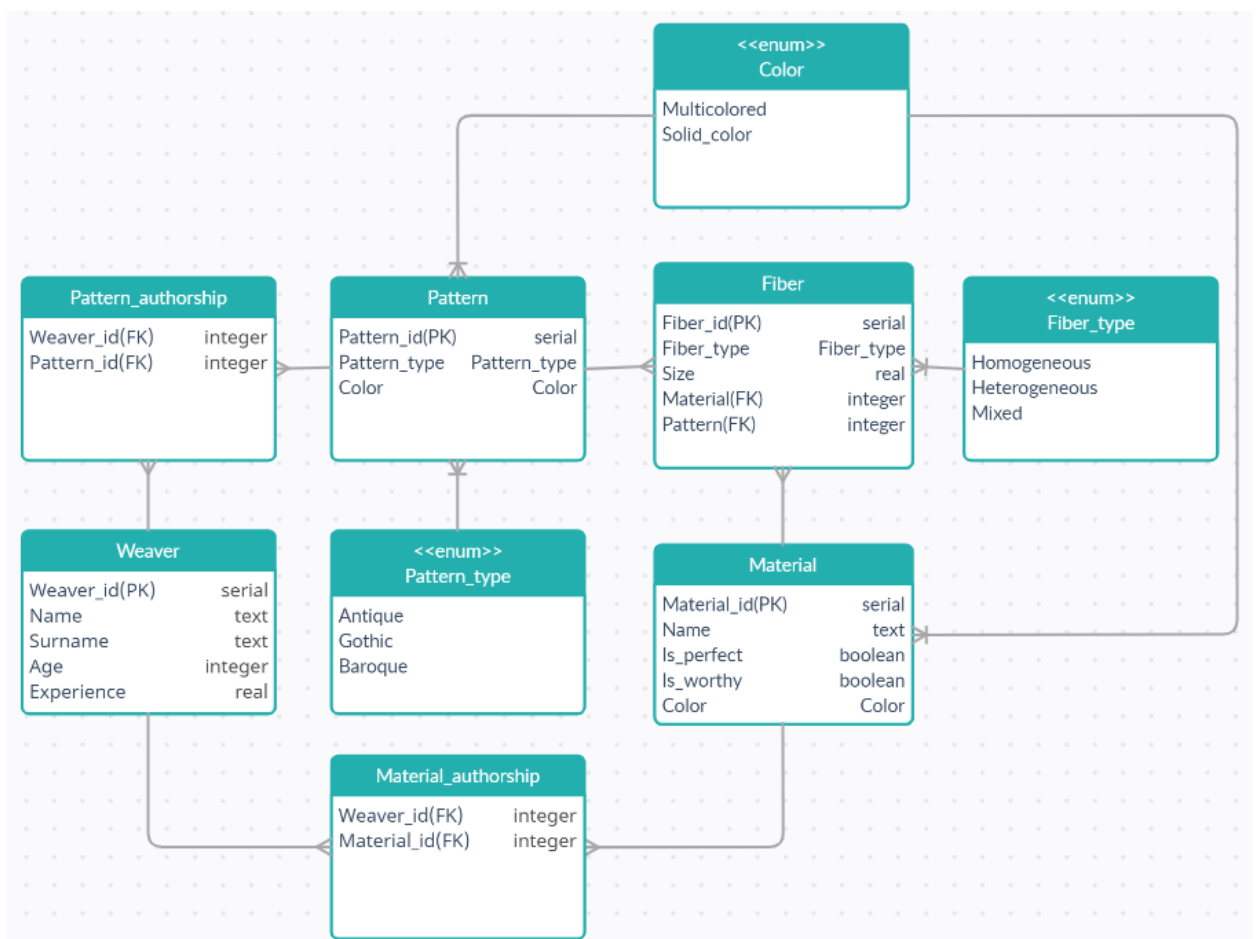
Группа: Р33301

Задание:

Для отношений, полученных при построении предметной области из лабораторной работы №1, выполните следующие действия:

- опишите функциональные зависимости для отношений полученной схемы (минимальное множество);
- приведите отношения в 3NF (как минимум). Постройте схему на основе NF (как минимум). Постройте схему на основе полученных отношений;
- опишите изменения в функциональных зависимостях, произошедшие после преобразования в 3NF (как минимум). Постройте схему на основе NF;
- преобразуйте отношения в BCNF. Докажите, что полученные отношения представлены в BCNF;
- какие денормализации будут полезны для вашей схемы? Приведите подробное описание;

Функциональные зависимости для отношений полученной схемы:



pattern_authorship	Ассоциативная сущность
pattern	pattern_id -> pattern_type pattern_id -> color
fiber	fiber_id -> fiber_type fiber_id -> size fiber_id -> material fiber_id -> pattern
weaver	weaver_id -> name weaver_id -> surname weaver_id -> age weaver_id -> experience
material	material_id -> name material_id -> is_perfect material_id -> is_worthy material_id -> color
material_authorship	Ассоциативная сущность

Отношения в 3NF:

Форма	Требование
1NF	<ul style="list-style-type: none"> В таблице не должно быть дублирующих строк В каждой ячейке таблицы хранится атомарное значение (одно не составное значение) В столбце хранятся данные одного типа Отсутствуют массивы и списки в любом виде
2NF	<ul style="list-style-type: none"> Таблица должна находиться в первой нормальной форме Таблица должна иметь ключ Все неключевые столбцы таблицы должны зависеть от полного ключа (в случае если он составной)
3NF	<ul style="list-style-type: none"> Требование третьей нормальной формы (3NF) заключается в том, чтобы в таблицах отсутствовала транзитивная зависимость. <i>Транзитивная зависимость</i> – это когда неключевые столбцы зависят от значений других неключевых столбцов.
BCNF	<ul style="list-style-type: none"> Таблица должна находиться в третьей нормальной форме. Ключевые атрибуты составного ключа не должны зависеть от неключевых атрибутов. (Часть составного первичного ключа не должна зависеть от неключевого столбца.)

Отношение	Схема в 3NF
pattern_authorship	
pattern	
fiber	
weaver	
material	
material_authorship	

Изменения в функциональных зависимостях, произошедшие после преобразования в 3NF:

Никаких изменений не произошло, поскольку, оказалось, что отношения изначально были в NF3.

Отношения в BCNF. Доказательство, что полученные отношения представлены в BCNF:

Для начала докажем, что отношения находятся в 3NF, т.к. это необходимое условие для того, чтобы они находились в BCNF (нормальной форме Бойса-Кодда).

Во-первых, в таблице не будет дублирующих строк, т.к. у каждого отношения есть свой РК и даже, если все остальные ячейки будут идентичными в некоторых строках, то id все равно останется уникальным. Так же, в каждой ячейке хранится только одно значение и в столбце хранятся только данные одного типа, при этом нет каких-либо массивов и списков.

Из этого следует, что отношения представлены в 1NF.

Во-вторых, все таблицы имеют ключ, а также неключевые столбцы зависят от полного ключа.

В следствии чего, можно сказать, что отношения представлены в 2NF.

В-третьих, все неключевые столбцы зависят только от ключевых столбцов.

А значит, отношения представлены в 3NF.

Вторым условием, доказывающим представленность данных в BCNF, является то, что часть составного первичного ключа не должна зависеть от столбца, не являющимся ключом. В моей базе данных есть только два составных ключа, помимо которых в соответствующих таблицах нет других полей, поэтому данное условие выполняется.

Следовательно, в моей базе данных отношения представлены в BCNF.

Какие денормализации будут полезны для вашей схемы? Приведите подробное описание:

Как мне кажется, в моей схеме денормализации не особо нужны, поскольку отношения изначально были представлены в BCNF; они, скорее всего, только ухудшат понимание схемы. Тем более в таком виде она соответствует некому “эталоноу”, что повышает производительность при работе с этой базой данных (отсутствует избыточность данных), также управление БД в данном случае более гибкое и удобное (для чего, собственно, и проводится нормализация).

Вывод:

Несмотря на то, что нормализация данных помогает избавиться от избыточности данных, что может повысить производительность при работе с БД, а также делает управление базой данных более гибким и удобным, главное помнить, что все хорошо в меру и не переборщить с этим. Так как если сделать слишком много декомпозиций и ненужных таблиц, то производительность только упадет.