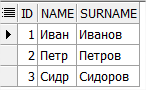
1. Нормальная форма – это некоторое требование, предъявляемое к бд, которое позволяет привести бд к некоторому стандарту.

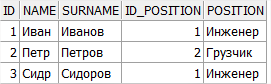
Существует 8 нормальных форм, на практике достаточно 3.

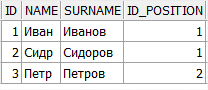
Первая нормальная форма соблюдается тогда, когда каждый атрибут таблицы атомарен (имеет одно значение), строки не повторяются.

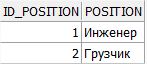
 - соответствует первой нормальной форме.

 - не соответствует первой нормально форме.

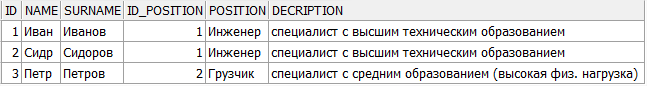
Вторая нормальная форма соблюдается тогда, когда выполняет первая нормальная форма и каждый атрибут, который не является первичным ключом полностью зависит от первичного ключа.

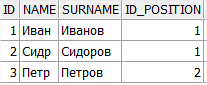
 - не находится во 2 нормальной форме

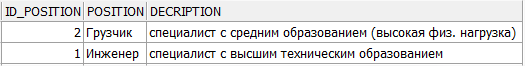


 - соответствует второй нормальной форме.

Третья нормальная форма соблюдается тогда, когда выполняется вторая нормальная форма и отсутствуют транзитивные зависимости (неключевые атрибуты зависят от других неключевых атрибутов).

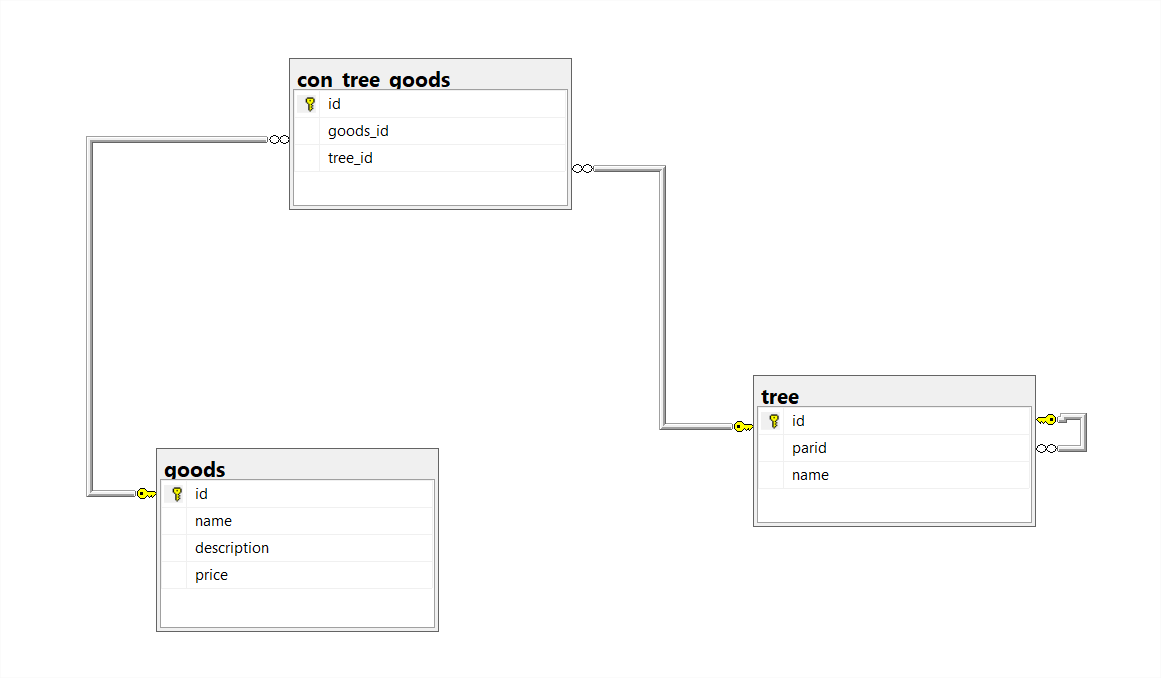
- не находится в 3 нормальной форме



 - соответствует третьей нормальной форме.

Третья усиленная нормальная форма (Бойса-Кодда) соблюдается тогда, когда выполняется третья нормальная форма и ключевые атрибуты не зависят от неключевых.





-- Таблица товаров

**CREATE** **TABLE** goods (**id** **INTEGER** **NOT** **NULL**, name **VARCHAR2** (100 **CHAR**), description **VARCHAR2** (250 **CHAR**), price **REAL**,

**CONSTRAINTS** g\_pk\_id **PRIMARY** **KEY** (**id**));

-- дерево групп, типов, видов

**CREATE** **TABLE** tree (**id** **INTEGER** **NOT** **NULL**, parid **INTEGER**, name **VARCHAR2** (100 **CHAR**),

**CONSTRAINTS** t\_pk\_id **PRIMARY** **KEY** (**id**));

**ALTER** **TABLE** tree

**ADD** **CONSTRAINT** t\_fk\_id **FOREIGN** **KEY** (parid)

**REFERENCES** tree (**id**);

-- таблица для связи "много ко многому"

**CREATE** **TABLE** con\_tree\_goods (**id** **INTEGER** **NOT** **NULL**, goods\_id **INTEGER**, tree\_id **INTEGER**,

**CONSTRAINTS** ctg\_pk\_id **PRIMARY** **KEY** (**id**),

**CONSTRAINTS** ctg\_fk\_goods\_id **FOREIGN** **KEY** (goods\_id)

**REFERENCES** goods (**id**),

**CONSTRAINTS** ctg\_fk\_tree\_id **FOREIGN** **KEY** (tree\_id)

**REFERENCES** tree (**id**));

**INSERT** **ALL**

**INTO** tree (**id**, name) **VALUES** (1, 'техника')

**INTO** tree **VALUES** (2, 1, 'Бытовая техника')

INTO tree VALUES (3, 2, 'Стиральные машины')

INTO tree VALUES (4, 2, 'Посудомойки')

INTO tree VALUES (5, 2, 'Пылесосы')

INTO tree (id, name) VALUES (6, 'Мебель')

INTO tree VALUES (7, 6, 'Кухонная мебель')

INTO tree VALUES (8, 6, 'Мебель для спальни')

INTO tree VALUES (9, 7, 'Столы')

INTO tree VALUES (10, 7, 'Стулья')

INTO tree VALUES (11, 8, 'Кровати')

SELECT \* FROM DUAL;

INSERT ALL

INTO goods VALUES (1, 'Indesit', 'С функцией отжима', 17000)

INTO goods VALUES (2, 'Samsung', 'Надежный производитель', 23000)

INTO goods VALUES (3, 'LG', 'Проверенно временем!', 15000)

INTO goods VALUES (4, 'Philips', 'Выгодная цена', 12000)

**INTO** goods **VALUES** (5, 'bosch', 'цена/качество', 16000)

**INTO** goods **VALUES** (6, 'Кресло ‘"Бюрократ"', 'Офисное', 4000)

INTO goods VALUES (7, 'Стул '"КАУСТ"', 'Офисное', 2300)

INTO goods VALUES (8, 'Кресло "RAZOR"', 'Игровое', 8000)

INTO goods VALUES (9, 'Стол "Райдер"', 'Большой', 6000)

**INTO** goods **VALUES** (10, 'Кровать "Орматек"', 'Двуспальная', 18000)

**SELECT** \* **FROM** DUAL;

**INSERT** **ALL**

**INTO** con\_tree\_goods **VALUES** (1, 1, 1)

**INTO** con\_tree\_goods **VALUES** (2, 1, 2)

**INTO** con\_tree\_goods **VALUES** (3, 1, 3)

**INTO** con\_tree\_goods **VALUES** (4, 2, 3)

**INTO** con\_tree\_goods **VALUES** (5, 2, 4)

**INTO** con\_tree\_goods **VALUES** (6, 2, 5)

**INTO** con\_tree\_goods **VALUES** (7, 2, 5)

**INTO** con\_tree\_goods **VALUES** (8, 3, 1)

**INTO** con\_tree\_goods **VALUES** (9, 3, 2)

**INTO** con\_tree\_goods **VALUES** (10, 3, 3)

**INTO** con\_tree\_goods **VALUES** (11, 3, 4)

**INTO** con\_tree\_goods **VALUES** (12, 4, 1)

**INTO** con\_tree\_goods **VALUES** (13, 4, 2)

**INTO** con\_tree\_goods **VALUES** (14, 4, 3)

**INTO** con\_tree\_goods **VALUES** (15, 5, 1)

**INTO** con\_tree\_goods **VALUES** (16, 5, 2)

**INTO** con\_tree\_goods **VALUES** (17, 5, 3)

**INTO** con\_tree\_goods **VALUES** (18, 6, 6)

**INTO** con\_tree\_goods **VALUES** (19, 6, 8)

**INTO** con\_tree\_goods **VALUES** (20, 6, 10)

**INTO** con\_tree\_goods **VALUES** (21, 7, 6)

**INTO** con\_tree\_goods **VALUES** (22, 7, 8)

**INTO** con\_tree\_goods **VALUES** (23, 7, 10)

**INTO** con\_tree\_goods **VALUES** (24, 8, 6)

**INTO** con\_tree\_goods **VALUES** (25, 8, 8)

**INTO** con\_tree\_goods **VALUES** (26, 8, 10)

**INTO** con\_tree\_goods **VALUES** (27, 9, 6)

**INTO** con\_tree\_goods **VALUES** (28, 9, 7)

**INTO** con\_tree\_goods **VALUES** (29, 9, 9)

**INTO** con\_tree\_goods **VALUES** (30, 10, 6)

**INTO** con\_tree\_goods **VALUES** (31, 10, 7)

**INTO** con\_tree\_goods **VALUES** (32, 10, 8)

**INTO** con\_tree\_goods **VALUES** (33, 10, 11)

**SELECT** \* **FROM** DUAL;

-- Товары, входящие в «Стиральные машины»

**SELECT** \*

**FROM** goods

**WHERE** goods.**id** **IN**

(**SELECT** goods\_id

**FROM** con\_tree\_goods

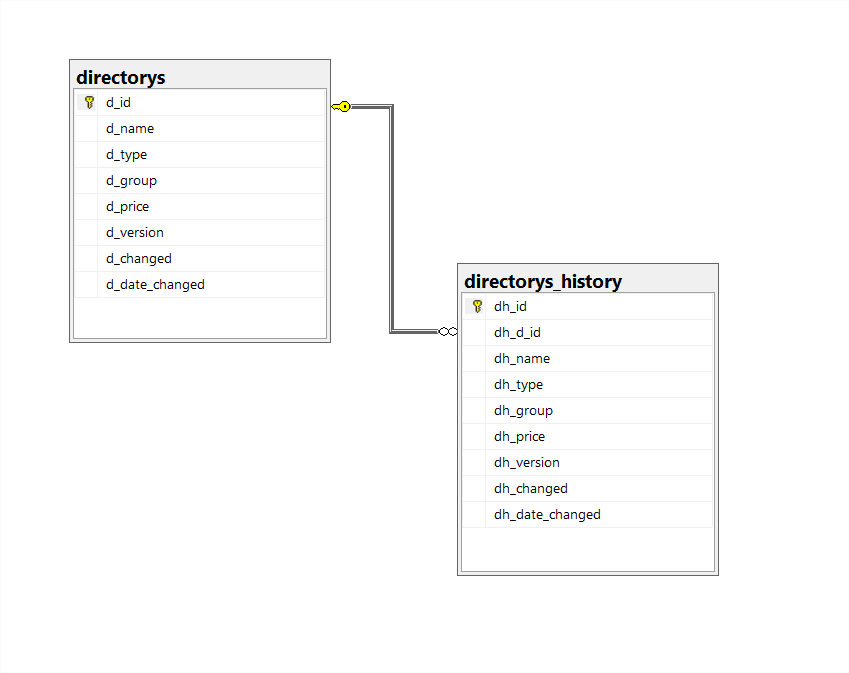
**WHERE** tree\_ID **IN**

(**SELECT** **ID**

**FROM** tree

**START** **WITH** tree.name = 'Стиральные машины'

**CONNECT** **BY** **PRIOR** tree.**ID** = tree.parid))

1. 

Создаем две таблицы одинаковой структуры: «справочник и история справочника». Добавляем в обе таблицы поля «Version» - счетчик. При изменении данных в таблице «directorys» срабатывает триггер, который увеличивает поле «Version», затем копирует данные из «directorys» в «directorys\_history»