

Раздел 8

Создание банка данных MSC.Mvision - форма и синтаксис

БАНК ДАННЫХ - ОСНОВЫ

- n Важнейшим моментом при создании системы управления базой данных является понимание структуры и содержания банка данных
- n Банк данных – совокупность иерархически структурированных данных
- n Все банки данных используют **схему** для описания иерархии

БАНК ДАННЫХ - ОСНОВЫ

- n Иерархические отношения и определения атрибутов (структура данных и тип) могут изменяться с помощью:
 - u **Перезагрузки данных с измененным определяющим и/или входным файлами**
 - u **DPI**
 - u **Builder (File/Builder/...)**
 - u **Текстового интерфейса Builder (mvbatch builder)**

БАНК ДАННЫХ - ОСНОВЫ

- n Банки данных могут содержать различные типы данных:
 - и Символы
 - и Вещественные и целые числа
 - и Рисунки (графики)
 - и Изображения

ОПРЕДЕЛЕНИЯ БАНКА ДАННЫХ

- n Атрибуты определяют ячейки, содержащие информацию из «генеалогии» записи о свойствах материала, а также непосредственно данные о материале

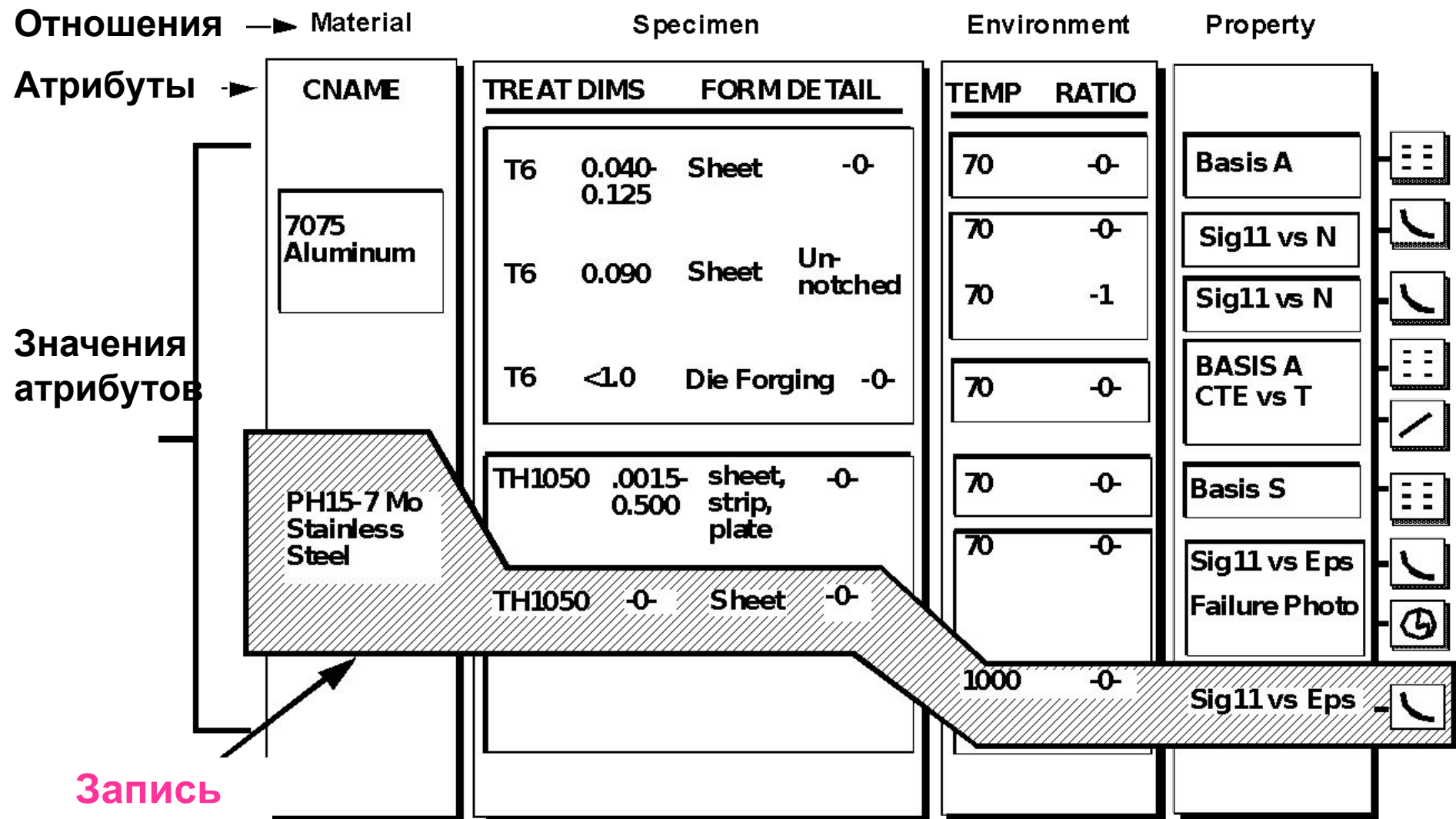
например: E11T, SIG11TvsEPS, CNAME, TREAT, и т.д.

- n Значение атрибута определяет фактическую часть данных, соответствующих конкретному атрибуту; например: 10 Мра
- n Отношения (relations) – непосредственно таблицы, объединяющие атрибуты и данные

ОПРЕДЕЛЕНИЯ БАНКА ДАННЫХ

- n Иерархия определяет структуру отношений
- n Уровень Свойств - самый низкий уровень в иерархии банка данных MSC.Mvision, который содержит только данные о материале (не содержит информацию для полной идентификации конкретного материала, т.е. генеалогию)
- n Запись — уникальная запись входа в банк данных по шагам, содержащая всю информацию о генеалогии и свойствах для данного материала

ПРИМЕР ИЕРАРХИИ БАНКА ДАННЫХ



Иерархия= Material, Specimen, Environment

СОВЕТЫ ПРИ ПРОЕКТИРОВАНИИ ОПРЕДЕЛЯЮЩЕГО ФАЙЛА БАНКА ДАННЫХ (DEFINE FILE)

- n Продумайте, как будет происходить обращение к данным
- n Например, нужно ли сохранять данные в текстовом формате или в виде графика?
- n Организуйте иерархию данных в том виде, в котором пользователи вероятнее всего будут их искать
- n Воспользуйтесь преимуществом терминологии и классификации, разработанной другими компаниями, существующими макетами электронных таблиц или информационными листками поставщика

СОВЕТЫ ПРИ ПРОЕКТИРОВАНИИ ОПРЕДЕЛЯЮЩЕГО ФАЙЛА БАНКА ДАННЫХ (DEFINE FILE)

- n Используйте рекомендуемые MSC имена атрибутов (*См. Документацию*).
- n Воспользуйтесь возможностью создания большого количества уровней отношений (до 32 уровней).

Старайтесь не располагать данные на слишком высоком уровне в иерархии отношений

Помните, что для поиска данных ваши пользователи могут использовать функции Query & Select.

- n Не используйте зарезервированные слова (*См. Документацию*)

ОПРЕДЕЛЯЮЩИЙ ФАЙЛ (DEFINE FILE) - ОСНОВЫ

- n Схема банка определяется создаваемым пользователем определяющим файлом в текстовом формате (*dbname.def*)
- n Определяющий файл содержит следующие записи:
 - u Определения атрибутов
 - | Имя, тип данных, описание, единицы, и т.д.
 - u Определения отношений
 - | Определяют один или несколько атрибутов для каждого отношения

ОПРЕДЕЛЯЮЩИЙ ФАЙЛ (DEFINE FILE) - ОСНОВЫ

- и Определение иерархии
 - | Определяется порядок уровней отношений, следует перед определениями отношений
- и Строки комментариев
 - | Это строки, начинающиеся с символа “\$”

ОБЗОР СТРУКТУРЫ БАНКА ДАННЫХ

- Отношения могут отображаться в иерархическом виде, как показано ниже

Уровни отношений (максимум 32)

Самый низкий уровень в иерархии отношений

Common P
Aluminum

Data Viewer

File Table XY Graphics View Tools

Select Property Set... PROPERTY

Attribute	Selection 1.1	Selection 1.2
DATE_EFF	90-11-01	90-11-01
DATE_MOD	91-10-15	91-10-15
US11T	80 ksi	78 ksi
US221	80 ksi	78 ksi
VS11T	73.0 ksi	71.0 ksi
VS22T	71.0 ksi	69.0 ksi
VS11C	72.0 ksi	70.0 ksi
VS22C	75.0 ksi	70.0 ksi

Атрибут Значение Единицы

Иерархия Банка данных

СОЗДАНИЕ ИЕРАРХИИ

Синтаксис:

HIE[RARCHY], ADD, *relation_1*, *relation_2*, ... *relation_n*, *relation_source*

Где:

relation_n список отношений

relation_source имя самого низкого уровня отношений

- и Используйте запятые или пробелы для разделения записей
- и При переносе строки используйте запятую в конце строки
- и **НЕ** включайте в уровни иерархии атрибуты, определяющие графики, табличные данные или изображения
- и Определение иерархии должно предшествовать определениям отношений
- и Последний уровень отношений в иерархии будет воспринят как **SOURCE**. Поэтому, Вы **ДОЛЖНЫ** определить этот уровень (SOURCE).

Отношения в структуре банка данных

ОПИСАНИЕ ОТНОШЕНИЙ

Синтаксис:

**REL[ATION], ADD, name_ attribute, attribute _1, attribute _2, ...
attribute _n**

Где:

name **назначенное отношеию имя**

attribute _n **один или несколько предварительно
определенных атрибутов**

- и Описанные отношения разделяются запятыми или пробелами
- и При переносе строки используйте запятую в конце строки
- и Убедитесь в правильности написания имен атрибутов!

ПРИМЕРЫ ОПРЕДЕЛЕНИЯ ОТНОШЕНИЙ

RELATION, ADD, MATERIAL, CNAME, UNS

RELATION, ADD, PROPERTY, E11T, YS11T, UE11T, YS22T

Будет выведена таблица характеристик материала на уровне Наборов свойств.

REL ADD ENVIRONMENT TEMP

REL, ADD, SIG11TvsEPS, SIG11TvsEPS (две последние записи должны быть одинаковыми)

Будет построен график SIG11TVSEPS (кривая деформирования)

REL, ADD, SOURCE, FIGURE, FIGURE_NAME, TABLE, TABLE_NAME, SCAN

ИСХОДНОЕ ОТНОШЕНИЕ (SOURCE RELATION)

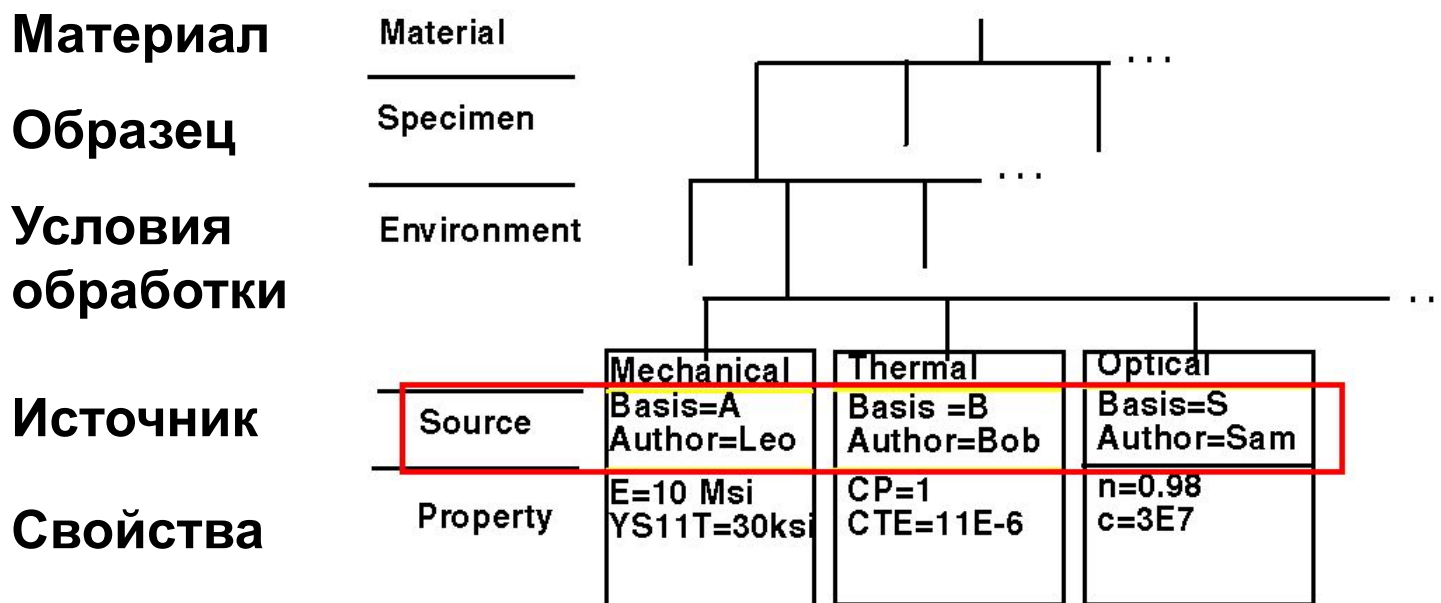
- n Используйте атрибуты группы SOURCE (источники данных), чтобы описать как, когда и откуда были получены данные.
 - u Примеры: название графика или таблицы, название лаборатории, наименование стандарта, дата испытаний, и т.д.
- n Данные SOURCE могут быть связаны с каждой уникальной записью данных (а также с каждой таблицей, графиком или изображением)

ИСХОДНОЕ ОТНОШЕНИЕ (SOURCE RELATION)

- n Данные SOURCE отображаются на уровне “Property Set” рядом с таблицами, рисунками и изображениями
 - и **Атрибуты SOURCE помогают описать содержание таблиц, графиков и изображений.**

ИСХОДНОЕ ОТНОШЕНИЕ (SOURCE RELATION)

- Отношение SOURCE не является максимальным из всех 32-х, но является последним отношением в выражении HIERARCHY при описании иерархии
- Для этого отношения рекомендуется использовать имя SOURCE, хотя можно присваивать и другие имена.



Атрибуты Данных

ОПИСАНИЕ АТТРИБУТОВ

Синтаксис:

ATT[RIBUTE], ADD, name, type, dimension_1, dimension _2, units, description, synonym, precision

Где:

- u name* *назначенное атрибуту имя*
- u type* *REAL, INTEGER, или CHARACTER*
- u dimension_1* *Значения зависят от типа данных*
- u dimension_2*
- u units* *единицы измерения*
- u description* *Описание атрибута*
- u synonym* *Дополнительное имя атрибута*
 (необязательно)
- u precision* *Точность значений атрибутов в заданных по умолчанию*
 единицах (необязательно)

ОПРЕДЕЛЕНИЕ АТТРИБУТОВ

- Используются запятые или пробелы для разделения записей
- При переносе строки используйте запятую в конце строки

РАЗМЕРНОСТЬ	РАЗМЕРНОСТЬ_1	РАЗМЕРНОСТЬ_2
Скаляр (Real, Integer)	1	1
Строка символов	максимум 2000	1
Массив	число столбцов	число строк
Изображение	3	0
График	3	0

ПРИМЕР ОПРЕДЕЛЕНИЯ АТТРИБУТОВ

ATTRIBUTE, ADD, E11T, REAL, 1, 1, Msi,

“Young’s Modulus in fiber direction in tension”, E_t, 1.0

ATTRIBUTE, ADD, CNAME, CHARACTER, 80, 1, -0-,

“Common material name”

(“Примечание: -0- указывает на нулевое (пустое) значение.”)

ATT ADD TEMP REAL 1 1 “deg F” “Test Temperature”

ATT ADD BOOK_NAME CHARACTER 720 1 -0- “Book name”

(Атрибут BOOK_NAME - обычно часть отношения SOURCE)

ТОЧНОСТЬ ЧИСЛОВЫХ ДАННЫХ

- Для типа данных REAL задается некоторая точность, до которой округляется значение атрибута при выводе

- Примеры:**

Атрибут	Rho (Плотность)	E11T (Модуль)
Точность	0.001	1.0
Значение	0.10123	28.51 Msi
Отображаемое значение	0.101	29 Msi

ТОЧНОСТЬ ЧИСЛОВЫХ ДАННЫХ

- n Загруженное в банк данных значение сохраняется независимо от точности
- n Любые последующие операции выполняются с сохраненным значением. Затем, полученное значение округляется (например после преобразования системы единиц)

ТОЧНОСТЬ ЧИСЛОВЫХ ДАННЫХ

- n Запросы работают с отображенными, а не сохраненными значениями
- n Если значения для конкретного атрибута получены из различных источников или относятся к широкому ряду материалов, высокая точность может быть нежелательна

ТОЧНОСТЬ ЧИСЛОВЫХ ДАННЫХ

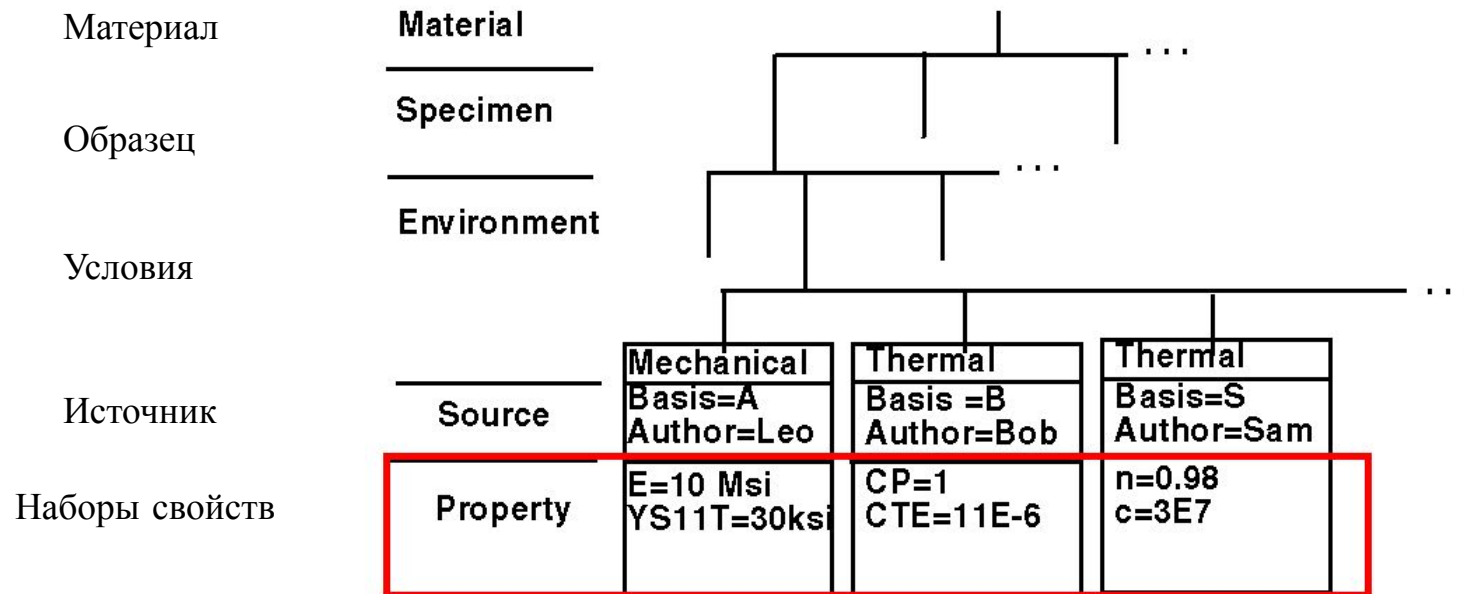
- n Если параметр точности опущен, MVISION автоматически определяет эффективную точность отображения в каждом конкретном случае
- n При использовании в запросе атрибутов вещественного типа, используйте относительные запросы или по крайней мере диапазон значений, который включает достаточный допуск, чтобы учесть округление
 - u Пример: $E11T > 0.8 \text{ AND } E11T < 11.0$

Уровень данных

(Databank “Property Sets” Level)

УРОВЕНЬ ДАННЫХ

- Самый нижний уровень банка данных
- Здесь хранятся все графики, таблицы и изображения (наборы свойств)
- Может содержать неограниченное число отношений



УРОВЕНЬ ДАННЫХ

- n Наборы свойств банка данных соответствует данным, отображаемым при просмотре в окне Data Viewer
- n Наборы свойств включают в себя следующие типы данных: вещественный (E11T), символьный (Basis), целочисленный, графики (SIG11TvsEPS), изображения (failure_photo), таблицы (датчики деформаций)
- n Все вещественные, символьные, и целочисленные данные представлены соответствующими атрибутами и входят в какую-либо таблицу-отношение (атрибуты разного типа могут находиться в одной таблице)
 - u Эта последняя группа отношений находится на самом низком уровне (уровне свойств) банка данных и не является частью общей иерархии банка

УРОВЕНЬ ДАННЫХ

- n Все графики, например (SIG11TvsEPS), изображения, и массивы автоматически относятся к уровню данных и представляют собой самостоятельную группу
 - u В результате, рисунки, изображения, и массивы обычно располагаются внизу определяющего файла (define file)
 - u Графики и изображения обычно группируются парами под одним атрибутом и **НЕ** являются частью общей иерархии банка
 - u Массивы обычно вызываются при помощи отношения свойств (на уровне данных), описание которого **НЕ** входит в команду определения иерархии HIE

ОПРЕДЕЛЕНИЕ ГРАФИКОВ

**ATTRIBUTE, ADD, <Y_name>vs<X_name>, REAL, 3, 0,
“X_units;Y_units”, “X_description;Y_description”**

**RELATION, ADD, <Y_name>vs<X_name>,
<Y_name>vs<X_name>**

- n НЕ указывайте отношение графика в команде HIERARCHY
- n Графики располагаются в наборах свойств (на уровне данных)
- n Имя отношения должно соответствовать имени атрибута
- n В имени атрибута следует использовать “vs”
- n Примеры: SIG11vsEPS, YS11TvsTEMP
- n Единицы X и Y и описания используются для маркировки осей при построении графика
- n Используйте атрибуты SOURCE “FIGURE” и “FIGURE_NAME”

ОПРЕДЕЛЕНИЕ МАССИВОВ (МАТРИЦ)

**ATT ADD strain_gauge REAL 3 1,
“micro in/in” “Strain Gauge Values” “_” 0.01**

- n** Массивы располагаются в наборах свойств (на уровне данных).
- n** Нет программного предела размера массива (ограничивается емкостью и эффективностью аппаратных средств)
- n** Массивы ДОЛЖНЫ быть полностью ЗАПОЛНЕННЫ
- n** В массивах допустимы только вещественные числа (кроме изображений)

ОПРЕДЕЛЕНИЕ ИЗОБРАЖЕНИЙ

ATTRIBUTE, ADD, SCAN, INTEGER, 3, 0

RELATION, ADD, SCAN, SCAN (the last two definitions must be the same)

- n НЕ указывайте отношение SCAN в выражении HIERARCHY
- n Определите и используйте атрибут FIGURE для указания имени и расположения внешнего графического файла (физически файл не добавляется в банк данных)
 - u Поэтому изображения не могут быть перемещены без обновления банка данных.
- n Изображения располагаются в наборах свойств (на уровне данных)
- n Данные изображений вводятся с помощью *входного файла (input file)* – подробнее о входном файле ниже

ОПРЕДЕЛЕНИЕ АТТРИБУТОВ ТЕКСТОВЫХ ФАЙЛОВ

ATTRIBUTE ADD FAB_CHARS CHAR 80 1 -0- "TEXT: Fabrication Characteristics"

- n** Внешние текстовые файлы обычно являются частью уровня данных; однако, они могут быть частью иерархии и, таким образом, появляться как объекты в списке материалов Material Browser
- n** Полные текстовые файлы - отдельные документы, которые не хранятся в банке данных MVISION: сохраняются только путь и имя файла (подобно изображениям)
- n** Слово " TEXT: " указывает, что атрибут является ссылкой на текстовый файл
- n** Размер файла не ограничен
- n** Размер строки (80 символов в вышеупомянутом примере) должен быть достаточно большим, чтобы обработать самое длинное ожидаемое имя файла