Referential Integrity and Composer

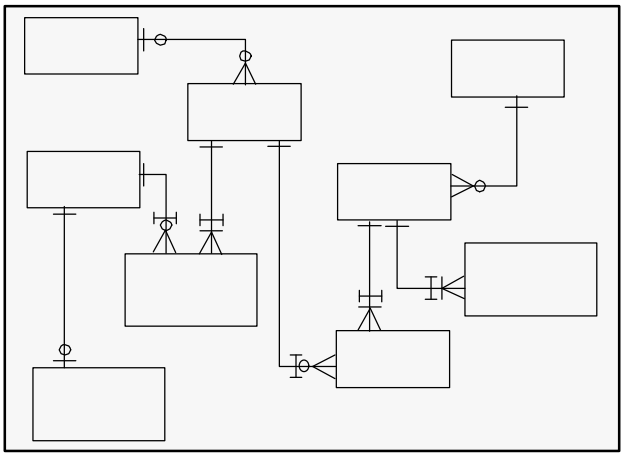
Excerpted From the Nims Courses: ***DB2™ For Composer/IEF™ Developers***
Oracle For Composer/IEF™ Developers
Technical Reviews For IEF™ Applications
Referential Integrity and the IEF™

Session 620

Introduction

- **RI is more than just DELETE rules**

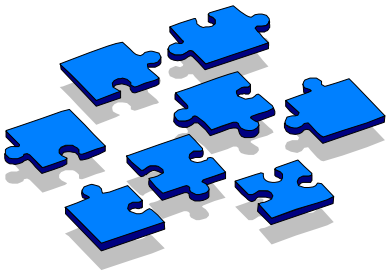
- **Other model issues and impacts**
  - **Acknowledged "holes" in RI enforcement**
  - **RI Action Block logic**
  - **DBMS vs IEF RI enforcement**
  - **Enforce Constraints flag**



Nims
The Strength of Information
Nims Associates, Inc.

RI Location in the IEF

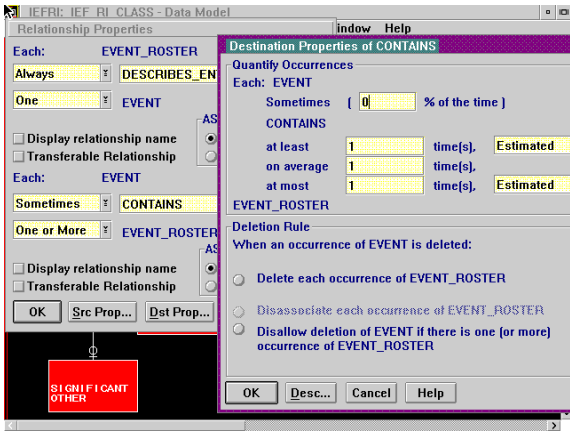
-
  -
  -
  -
  -
  -
  -



Nims The Strength of Information
Nims Associates, Inc.

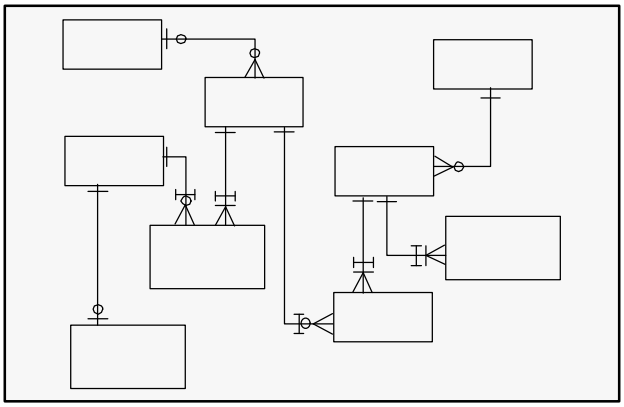
RI Rules in E/R Model

- **DELETE rules are specified via relationship properties**
  - **Src Prop and Dst Prop push-buttons**
  - **Invalid settings a**



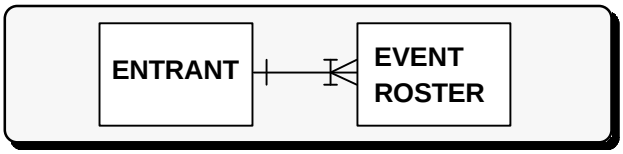
Default Delete RI Rules

- **DELETE Parent**
  - **Mandatory parent (BRACKET) - CASCADE to child**
  - **Optional parent (SPONSOR) - DISASS OCIATE (NULLify) child**
- **DELETE Child**
  - **Mandatory child (EMERGENCY CONTACT) - PENDANT to parent**
  - **Optional child (EVENT) - DISASSOCIATE from parent**



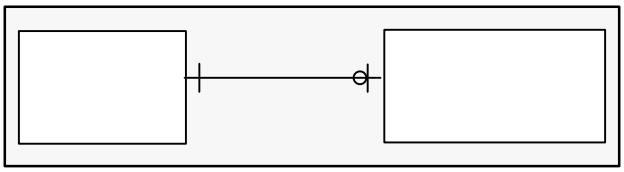
Nims
The Strength of Information
Nims Associates, Inc.

RI Issues From Relationship Cardinality



- **Fully mandatory relationship - child must be CREATEd at the same time as parent**
  - **INSERT RESTRICT**
- **IEF does not enforce this rule at run-time- Action Blocks can CREATE a parent with no child**
  - **Consistency check only validates existence of code, not logic flow**
  - **Leaves an integrity hole**
- **Solution - Ensure each CREATE parent AB for this type of relationship also CREATEs mandatory child**

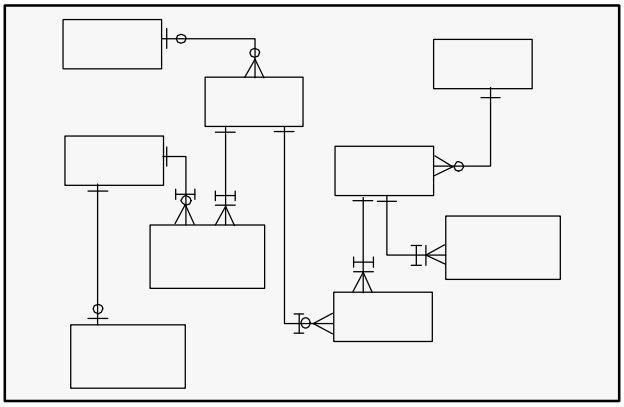
RI Issues From Relationship Cardinality



- **1:1 non-identifying - parent can only have one child**
- **IEF does not prevent 1:1 (non-identifying) from becoming 1:M**
- **Solution - Define child's foreign-key index as UNIQUE in TD**

RI Issues From Relationship Cardinality

- **Foreign key supporting any mandatory-one, non-identifying relationship is defined as NULLable (such as BRACKET to ENTRANT)**
  - ❑ **This foreign key should never contain NULLs**
  - ❑ **Could contain NULLs from non-IEF generated code, ad-hoc access, LOAD utility, improperly coded IEF code**
- **Consider changing foreign key definition in TD to NOT NULL**

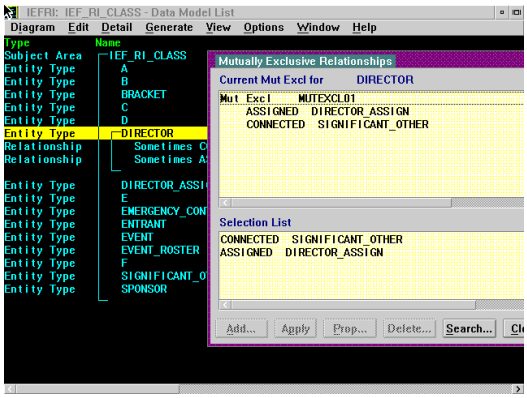


RI - Mutual Exclusivity

-

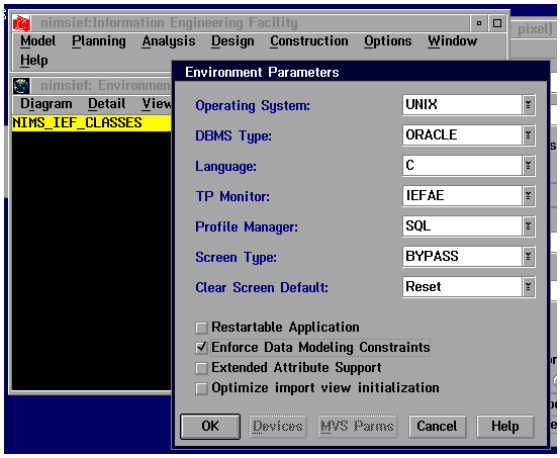
- **IEF does not enforce**

- **Solution - Any Action Block that CREATEs one dependent entity type must first check for the existence of the other dependent entity type(s)**



- **Construction flag to help enforce RI constraints not normally enforced by IEF**

- **If constructed with flag on, a constraint failure will result in a runtime err**

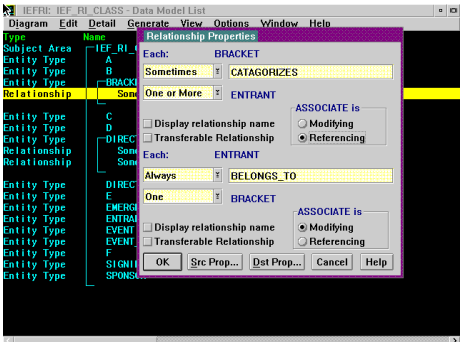


- **If set, construction will generate code to enforce four constraints**
  - **Prevent 1:1 from becoming 1:M**
  - **Enforce mutually exclusive relationships**
  - **Enforce fully-mandatory relationships**
  - **Prevent "quiet" disassociate**

- **Implications:**
  - **Action diagrams should contain this logic**
  - **Turn flag on to test logic in action diagrams (assumes complete test plan)**
  - **Turn flag off when generating after unit test**

- **Errors found at execution time will cause a runtime error**

- **Relationships defined with "referencing" have a potential RI hole**
  - **A parent (BRACKET) can be deleted at the same time a dependent (ENTRANT) is being associated to that parent if:**
    -
    -
    -

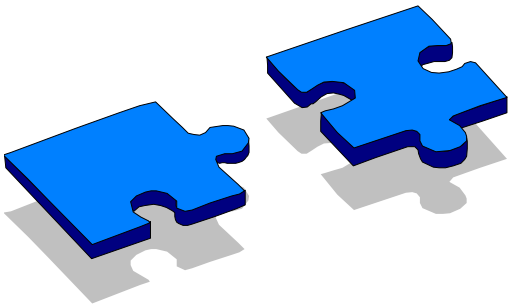


```
+-
| +- READ bracket
| |      WHERE DESIRED bracket code IS EQUAL TO import
| |            bracket code
| +- WHEN successful
| | +- CREATE entrant
| | | ASSOCIATE  WITH bracket WHICH categorizes IT
| | | SET ssn TO import entrant ssn
| | |   • • •
| | +- WHEN successful
| | +- WHEN already exists
| | +--
| +- WHEN not found
| | EXIT STATE IS bracket_nf
| +--
+--
```

```
EXEC SQL SELECT     BRACKET01."CODE"
           INTO     :CODE-001EF
           FROM     "BRACKET"                BRACKET01
           WHERE    BRACKET01."CODE" = :CODE-001TP
END-EXEC
```

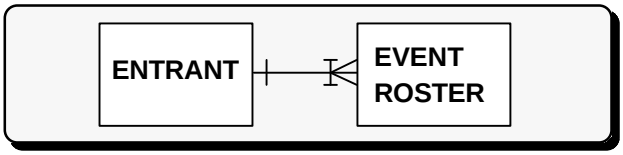
```
 EXEC SQL INSERT INTO "ENTRANT"
          (  "SSN",   "NAME",   "BIRTH_DATE",
            "SEX",   "ADDRESS",   FK_BRACKETCODE")
            VALUES (   :SSN-005EN,    :NAME-007EN,
           :BIRTH-DATE-009EN,   :SEX-011EN,    :ADDRESS-013EN,
         :FK-BRACKETCO-015EN)
 END-EXEC
```

- **Certain IEF-supported RI rules possibly <u>need supporting code</u> in action blocks**
  - ❑ **INSERT PARENT RESTRICT (fully-mandatory)**
  - ❑ **DELETE PARENT RESTRICT**
  - ❑ **DELETE PARENT CASCADE**
  - ❑ **DELETE CHILD PENDANT**
  - ❑ **DELETE CHILD RESTRICT**
  - ❑ **MUTUAL EXCLUSIVITY**



Nims The Strength of Information
Nims Associates, Inc.

- **DELETE PARENT RESTRICT**
  - **Specified via relationship properties**



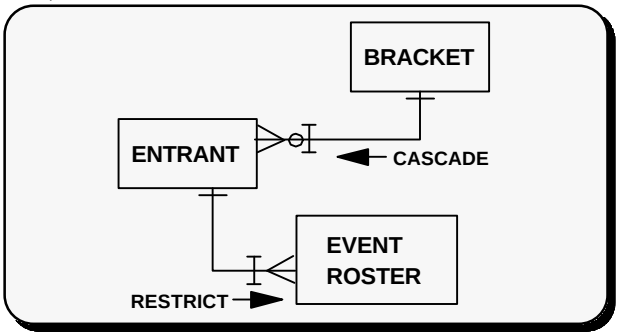
  - **Must also be coded in Action Block**
  - **If READ for existence of dependent is not done and dependent exists - user experiences fatal error**
    - **IEF provides no exception states for DELETEs - they are expected to succeed**

```
+-
| +- READ entrant
| |      WHERE DESIRED entrant ssn IS EQUAL TO import
| |            entrant ssn
| | += READ EACH event roster
| | |    WHERE DESIRED event roster sign_up_for CURRENT
| | |         event
| | |  EXIT STATE IS dependent _f
|<- - - ESCAPE
| +- WHEN successful
| | DELETE entrant
| +--
+--
```

RI Rule Support in Action Blocks

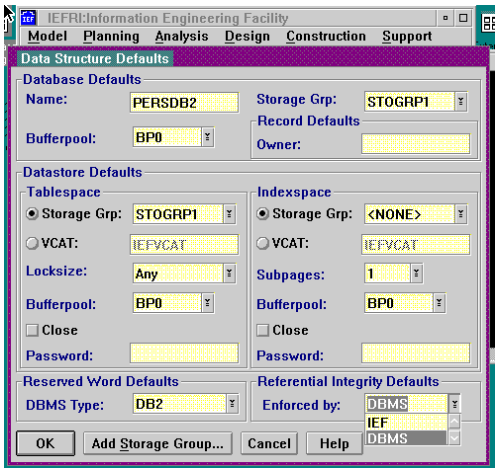
- **Example - DELETE PARENT CASCADE with downstream RESTRICT**

```
| +- READ bracket
| |      WHERE DESIRED bracket code IS EQUAL TO import bracket code
| +- WHEN successful
| |
| | += READ EACH event_roster
| | |      WHERE DESIRED event_roster contains SOME entrant
| | |      AND THAT entrant belongs_to SOME bracket
| | |      AND THAT bracket code IS EQUAL TO import bracket code
| | |
| | | NOTE Child found for a downstream restricted RI   rule
| | |
| | | EXIT STATE IS event_roster_fnd
|<------ESCAPE
| | +--
| | DELETE bracket
| +- WHEN not found
| | EXIT STATE IS bracket_nf
| +--
```



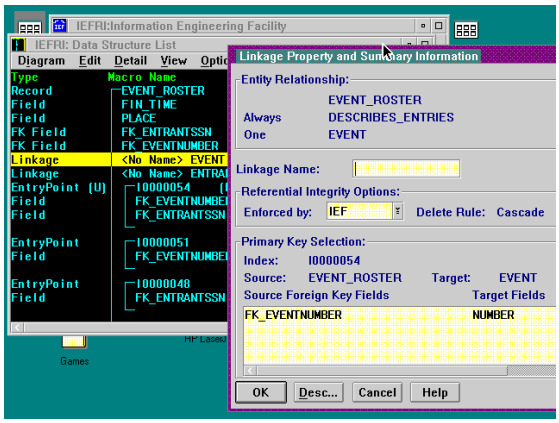
RI Enforcement Mechanism

- **Data structure default for RI enforcement needs to be set before transformation**
  - ❑ **DBMS - will generate a mix of IEF and DBMS support**
  - ❑ **IEF - will generate all RI support via RI triggers**
  - ❑ **Can possibly be changed on a relationship by relationship basi**



RI Enforcement Mechanism

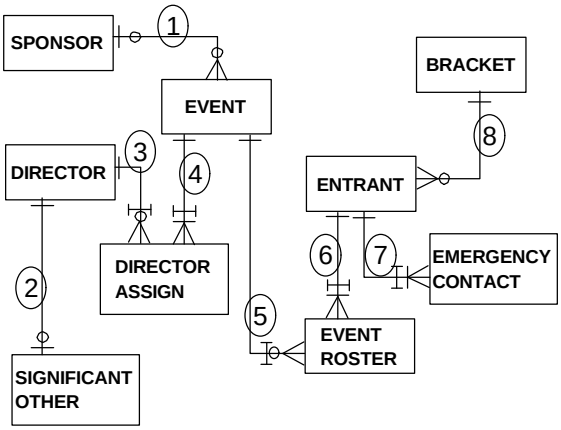
- **LINKAGE in Data Structure List contains physical RI rule properties**
  - ❑ **Resides in dependent "Record" pointing to parent**
  - ❑ **Specifies rule**
  - ❑ **Describes enforcement mechanism**
  - ❑ **<u>Only one</u> LINKAGE to support <u>both</u> DELETE rules (parent and dependent)**



Nims
The Strength of Information
Nims Associates, Inc.

RI Enforcement Implementation

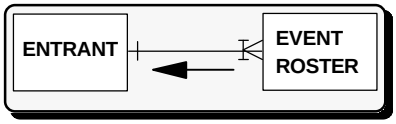
- **Given IEF defaults DELETE rules and DSD default of DBMS enforcement, what enforcement mechanism will be used for the following relationships?**



RI Enforcement Implementation

- **Relational DBMS's do not support some of IEF's RI rules.**
  - **If the relationship contains <u>one rule</u> the DBMS does <u>not</u> enforce, <u>all rule</u> enforcement on that relationship is IEF**

- **DB2 does not support**
  - **Parent DELETE NULLify or RESTRICT on a self-referencing relationship**
  - **Parent DELETE CASCADE through a complete cycle**
  - **Parent DELETE NULLify on parallel paths to the same dependent entity type**



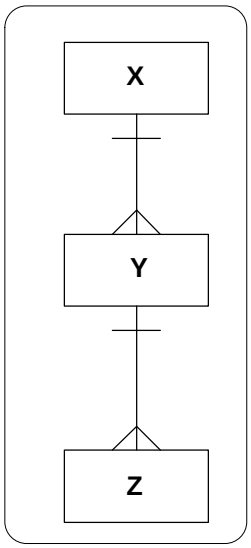
  - **<u>Dependent</u> DELETE RESTRICT**
  - **<u>Dependent</u> DELETE PENDANT**
  - **Above are the only DELETE options for dependents in fully-mandatory relationships - Therefore, fully mandatory relationships <u>cannot</u> be enforced by the DBMS**

RI Enforcement Implementation

- **Oracle does not support**
  - **Parent and child in different databases**
  - **Parent DELETE NULLify rule**
  - **Dependent DELETE RESTRICT**
  - **Dependent DELETE PENDANT**
  - **Above two are the only DELETE options for dependents in fully-mandatory relationships - Therefore, fully mandatory relationships cannot be enforced by the DBMS**

RI Enforcement Implementation

- **CASCADE relationships have impact on one another**



- **If this PARENT CASCADE**

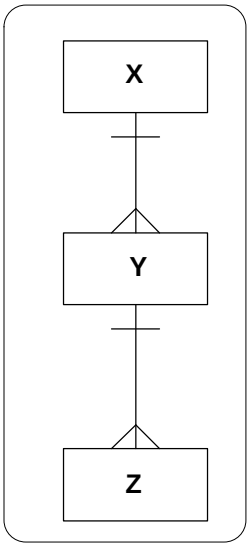
- **this relationship must be**

RI Enforcement Implementation

- **CASCADE Relationships have impact on one another**

- **this PARENT CASCADE must**

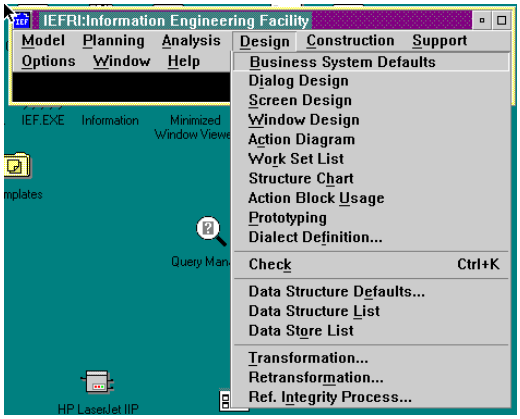
- **If this relationship is IEF-enforced**



RI Process

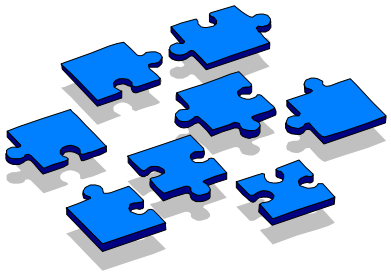
- **Function used to synchronize the logical RI rules with the physical RI rules and set enforcement to default**

- **Run whenever:**
  - ❑ **Logical RI rules change (can be done linkage by linkage as well)**
  - ❑ **Retransformation (automatically done in 5.3)**
  - ❑ **Reimplementation of entity type**



RI Location in the IEF

-
  -
  -
  -
  -
  -
  -



Nims The Strength of Information
Nims Associates, Inc.