

SQL Standard DDL Enhances Application Development with CA IDMS™ and Popular Tools

Joe Bedell
CA Technologies

IUA/CA IDMS™ Technical Conference
May 16-20, 2016



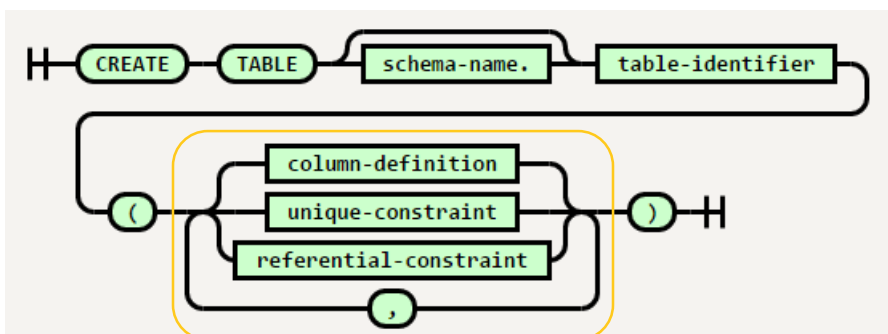
Abstract

- Users need to enhance developer productivity as they leverage and extend their investment in CA IDMS. Hibernate is a popular open source object to relational mapping framework for developing Java applications that access and store objects in relational databases. CA Test Data Manager (TDM), formerly known as Grid Tools, generates sophisticated sets of test data from relational databases for robust application testing. This session shows how CA IDMS 19.0 SQL DDL enhancements and CA IDMS Server can improve developer productivity by using Hibernate schema generation to create relational databases that provides persistence for Java objects, and by using CA TDM to create test databases based on real user data.

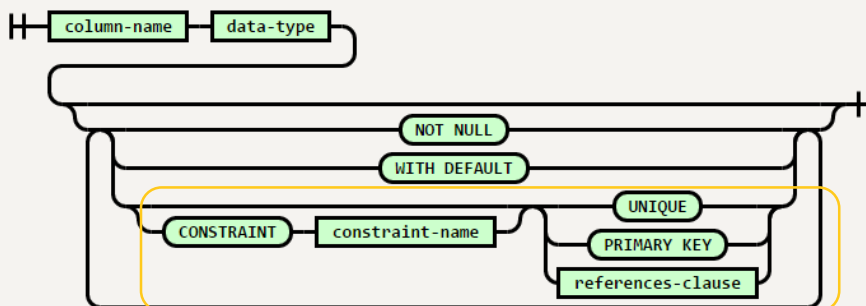
Agenda

- Standard DDL Enhancements
- Hibernate Schema Generation
- CA Test Data Manager

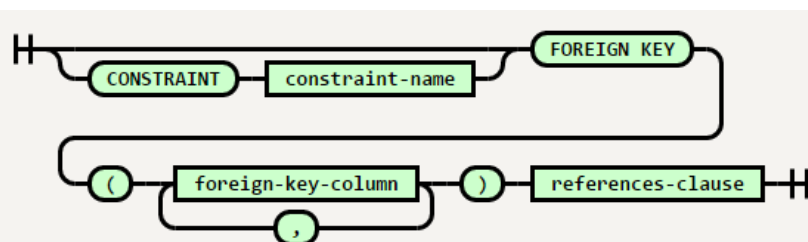
Create Table DDL



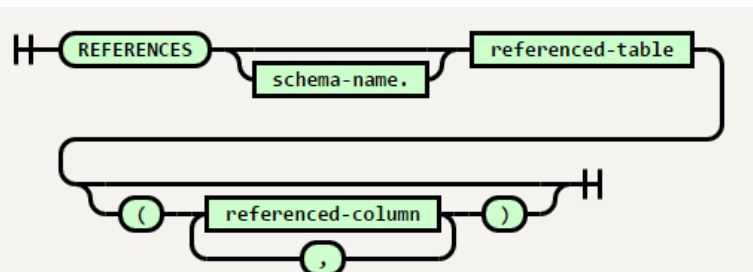
Column Definition



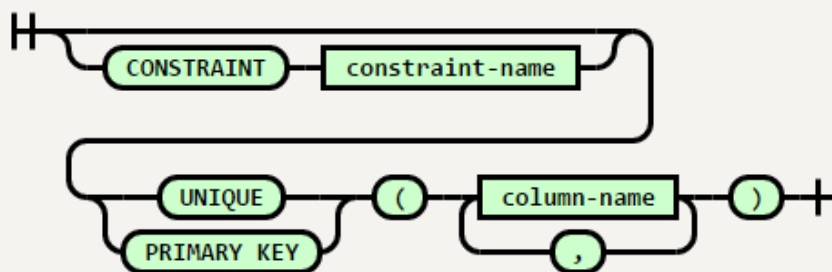
Referential Constraint



references-clause



Unique Constraint



Hibernate Schema Generation

What is object-relational mapping?

- From wikipedia.org:
 - “a programming technique for converting data between incompatible type systems in object oriented programming languages”
 - “virtual object database” used within the programming language
- Persistence
 - Objects stored, somewhere
 - Serialization
 - Database
- Reverse Engineering
- Schema Generation

9

IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.



Object-relational mapping concepts

Object (Java)

- Class
- Object
- Attribute
- Relationship

Relational (SQL)

- Table
- Row
- Column
- Referential constraint

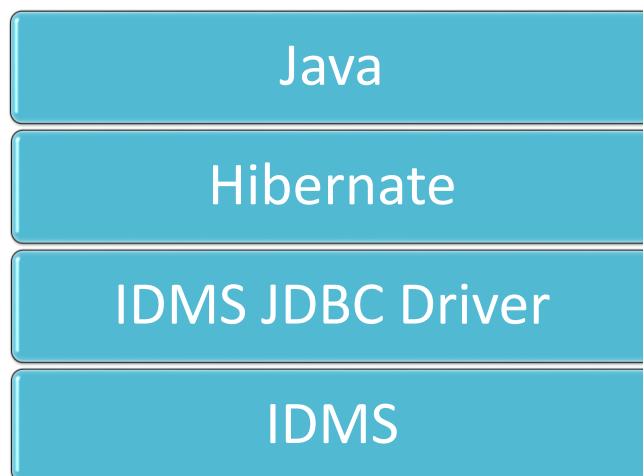
10

IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.



Hibernate architecture



11 IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.



Schema generation

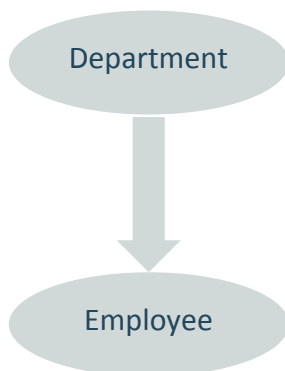
- Automatically generates DB schema from objects
- Most useful for prototyping DB
- Physical tuning always manual
- DBA should do final design

12 IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.



POJO: Plain Old Java Objects



Plain Old Java Employee

```

public class Employee {

    private String employeeSSN;
    private String employeeName;
    private String employeeTitle;
    private Dept employeeDept;
    private int employeeSalary;

    public Employee(String employeeName, String employeeTitle, String employeeSSN,
        Dept employeeDept, int employeeSalary)
    {
        this.employeeName = employeeName;
        this.employeeTitle = employeeTitle;
        this.employeeSSN = employeeSSN;
        this.employeeDept = employeeDept;
        this.employeeSalary = employeeSalary;
    }

    public int getEmployeeSalary() {
        return employeeSalary;
    }

    public void setEmployeeSalary(int employeeSalary) {
        this.employeeSalary = employeeSalary;
    }
    //more getter/setters
  
```

Plain Old Java Department

```
public class Dept {

    private long deptId;
    private String deptName;
    private Set<Employee> deptEmployeeNumbers = new HashSet<Employee>(0);

    public Dept (String deptName, int deptId)
    {
        this.deptName = deptName;
        this.deptEmployee = new HashSet<Employee>(0);
        this.deptId = deptId;
    }

    public long getDeptId() {
        return this.deptId;
    }

    public void setDeptId(long deptId) {
        this.deptId = deptId;
    }
}
```

Add Annotations to Employee

```
@Entity
@Table(name = "EMPLOYEE")
public class Employee {
    @Id
    @Column(name = "SSN", length=15)
    private String employeeSSN;

    @Column(name = "NAME", length=10)
    private String employeeName;

    @Column(name = "TITLE", length=15)
    private String employeeTitle;

    @ManyToOne(optional = false)
    @JoinColumn(name = "DEPT_ID", nullable = false)
    private Dept employeeDept;

    @Column(name = "SALARY", nullable = false, length=15)
    private int employeeSalary;
```


Add Annotations to Department

```
@Entity
@Table(name = "DEPT")
public class Dept {
    @Id
    @Column(name = "DEPT_ID")
    private long deptId;

    @Column(name = "DEPT_NAME", nullable = false, length = 100)
    private String deptName;

    @OneToMany(cascade = CascadeType.ALL, mappedBy =
        "employeeDept")
    private Set<Employee> deptEmployee = new HashSet<Employee>(0);
```

Persisting Object

```
Session session = HibernateUtil.getSessionFactory().openSession();
Transaction transaction = session.beginTransaction();

//Create a Department
Dept devDept = new Dept("Mainframe", 1);

//Add Employees
devDept.addEmployee("James", "Scrum Master", "111223333", 50000);
...

//Save the department
session.save(devDept);

//flush the session
transaction.commit();

//close the connection
session.close();
```

DDL to clean up

```
alter table IUA.EMPLOYEE
  drop constraint FK75C8D6AEFF9D0229;

drop table IUA.DEPT cascade;

drop table IUA.EMPLOYEE cascade;
```

Create tables

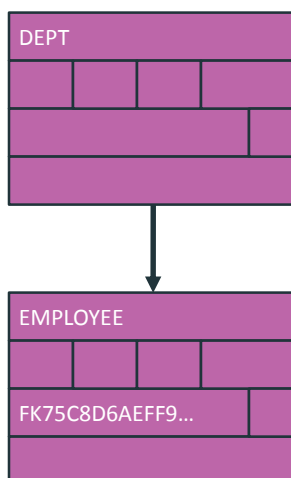
```
create table IUA.DEPT (
  DEPT_ID integer not null,
  DEPT_NAME varchar(100) not null,
  primary key (DEPT_ID)
);

create table IUA.EMPLOYEE (
  SSN varchar(15) not null,
  NAME varchar(20),
  TITLE varchar(255),
  SALARY integer not null,
  DEPT_ID integer not null,
  primary key (SSN)
);
```

Add Constraint

```
alter table IUA.EMPLOYEE
  add constraint FK75C8D6AEFF9D0229
  foreign key (DEPT_ID)
  references IUA.DEPT
```

Database representation



Create Department

```
insert
into
    IUA.DEPT
    (DEPT_NAME, DEPT_ID)
values
    (?, ?)
```

10:14:42 DEBUG StringType:80 - binding 'Mainframe' to parameter: 1
10:14:43 DEBUG LongType:80 - binding '1' to parameter: 2

Create Employees

```
insert
into
    IUA.EMPLOYEE
    (NAME, TITLE, DEPT_ID, SALARY, SSN)
values
    (?, ?, ?, ?, ?)
```

10:14:43 DEBUG StringType:80 - binding 'Mary' to parameter: 1
10:14:43 DEBUG StringType:80 - binding 'Product Owner' to parameter: 2
10:14:43 DEBUG LongType:80 - binding '1' to parameter: 3
10:14:43 DEBUG IntegerType:80 - binding '85000' to parameter: 4
10:14:43 DEBUG StringType:80 - binding '333445555' to parameter: 5

Add new Employee

```
//Find Department based on ID  
Dept devDept= (Dept) session.get(Dept.class, 1);
```

```
select  
    dept0_.DEPT_ID as DEPT1_0_0_,  
    dept0_.DEPT_NAME as DEPT2_0_0_  
from  
    IUA.DEPT dept0_  
where  
    dept0_.DEPT_ID=?
```

Add new Employee

```
//Add new employee to the Department  
devDept.addEmployee("Joe", "Developer", "123456789", 199);
```

```
select  
    deptemploy0_.DEPT_ID as DEPT5_1_,  
    deptemploy0_.SSN as SSN1_,  
    deptemploy0_.SSN as SSN1_0_,  
    deptemploy0_.NAME as NAME1_0_,  
    deptemploy0_.TITLE as TITLE1_0_,  
    deptemploy0_.DEPT_ID as DEPT5_1_0_,  
    deptemploy0_.SALARY as SALARY1_0_  
from  
    IUA.EMPLOYEE deptemploy0_  
where  
    deptemploy0_.DEPT_ID=?
```

Returns all of the Employees in the department with ID=1.

Add new Employee

```
//Save the department
session.save(devDept);

//close the transaction
transaction.commit();

insert
into
    IUA.EMPLOYEE
    (NAME, TITLE, DEPT_ID,
SALARY, SSN)
values
    (?, ?, ?, ?, ?)
```

27

IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.



Modify Employee

```
//Find employee based on SSN
Employee e1 = (Employee) session.get(Employee.class, "111223333");

select
    employee0_.SSN as SSN1_1_,
    employee0_.NAME as NAME1_1_,
    employee0_.TITLE as TITLE1_1_,
    employee0_.DEPT_ID as DEPT5_1_1_,
    employee0_.SALARY as SALARY1_1_,
    dept1_.DEPT_ID as DEPT1_0_0_,
    dept1_.DEPT_NAME as DEPT2_0_0_
from
    IUA.EMPLOYEE employee0_
inner join
    IUA.DEPT dept1_
        on employee0_.DEPT_ID=dept1_.DEPT_ID
where
    employee0_.SSN=?

09:14:18,956 DEBUG StringType:80 - binding '111223333' to parameter: 1
```

28

IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.



Modify Employee

```
09:14:19,039 DEBUG LongType:122 - returning '1' as column:
DEPT1_0_0_
09:14:19,042 DEBUG StringType:122 - returning 'Mainframe' as
column: DEPT2_0_0_
09:14:19,043 DEBUG StringType:122 - returning 'James' as column:
NAME1_1_
09:14:19,043 DEBUG StringType:122 - returning 'Scrum Master' as
column: TITLE1_1_
09:14:19,044 DEBUG LongType:122 - returning '1' as column:
DEPT5_1_1_
09:14:19,044 DEBUG IntegerType:122 - returning '100000' as column:
SALARY1_1_
```

Modify Employee

```
int empSalary = e1.getEmployeeSalary();
e1.setEmployeeSalary(empSalary*2); //double salary

//Save the new Employee
session.save(e1);

update
  IUA.EMPLOYEE
set
  NAME=?, TITLE=?,
  DEPT_ID=?,
  SALARY=?
where
  SSN=?
```

Delete Employee

```
//Find employee based on SSN
Employee e1 = (Employee) session.get(Employee.class, "123456789");

//Delete employee
session.delete(e1);

//close the transaction
transaction.commit();

delete
  from
    IUA.EMPLOYEE
  where
    SSN=?
```

11:54:43 DEBUG StringType:80 - binding '123456789' to parameter: 1

Hibernate Summary

- Java
 - Created classes with attributes to map to SQL tables
 - Created objects to become rows
 - Defined relationships
- Hibernate
 - Examined the generated code
 - Inserted, modified, and deleted data

CA Test Data Manager

What is CA Test Data Manager (TDM)?

- New CA product
- One-stop-shop for all of your data needs
- Create comprehensive test data

CA Test Data Manager Capabilities

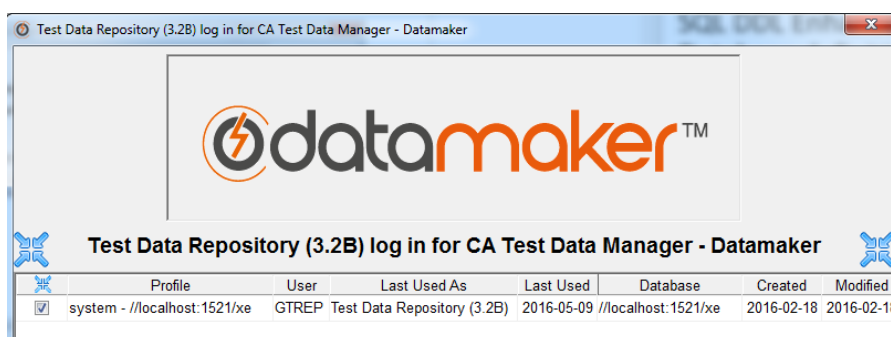
- Discover your data
- Sample and profile your data
- Subset production data for testing
- Mask production or data subset for testing
- Visualize test data coverage and identify gaps
- Generate synthetic data to fill test coverage gaps
- Build a test data repository and configure a data request and reservation system

35 IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.



Local database to use



36 IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.



Uses ODBC data source to connect to IDMS systems

	Profile	DBMS	Database	User	Last Used As
<input checked="" type="checkbox"/>	SYSQAS2	CA IDMS (ODBC)	SYSQAS2-DEBUG	joe	Data Target
<input type="checkbox"/>	SYSQAS1	CA IDMS (ODBC)	SQL_Server	bedjo05	Data Source

User DSN

Name	Driver
ODBCQA	CA IDMS
OINX	CA IDMS
SYS246	CA IDMS
SYS246-USI29CME	CA IDMS
SYSQAS1	CA IDMS
SYSQAS2-DEBUG	CA IDMS
SYSQAS7	CA IDMS

System DSN

Name	Driver
ERwin	DataDir
ERwin_Current	DataDir
SQL_Server	CA IDMS
Xtreme Sample Database 2008	Microsc

37 IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.



Visualize your data

Registering tables

Register Tables from Data Target into Project: IDMSDemo Version: base

Current Context - Project: IDMSDemo Version: base

- AA
 - Tables
 - Views
- AA3
- AA4
- CAW2013
- CJDEMP
- CTS
- DBCRSQLS
- DEMO
- DEMOEMPL
- DEMOEMPL2
- DEMOPROJ
- DUD
- EMPSQL
- EMPWITH
 - Tables
 - Views
- EMPWOUT

13 Tables for schema/user EMPWITH

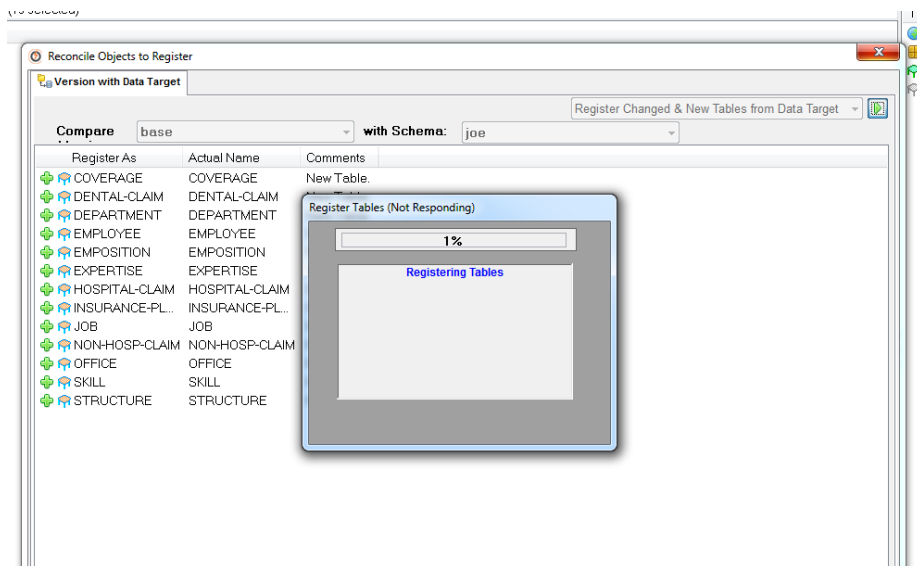
Name
COVERAGE
DENTAL-CLAIM
DEPARTMENT
EMPLOYEE
EMPOSITION
EXPERTISE
HOSPITAL-CLAIM
INSURANCE-PLAN
JOB
NON-HOSP-CLAIM
OFFICE
SKILL
STRUCTURE

38 IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.



Register your tables

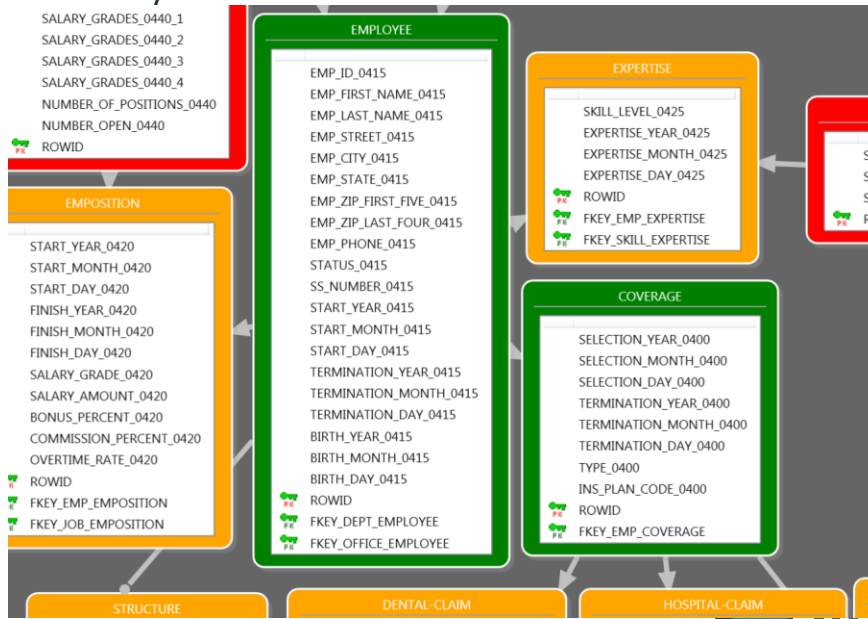


39 IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.



Visualize your Schema



40 IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.



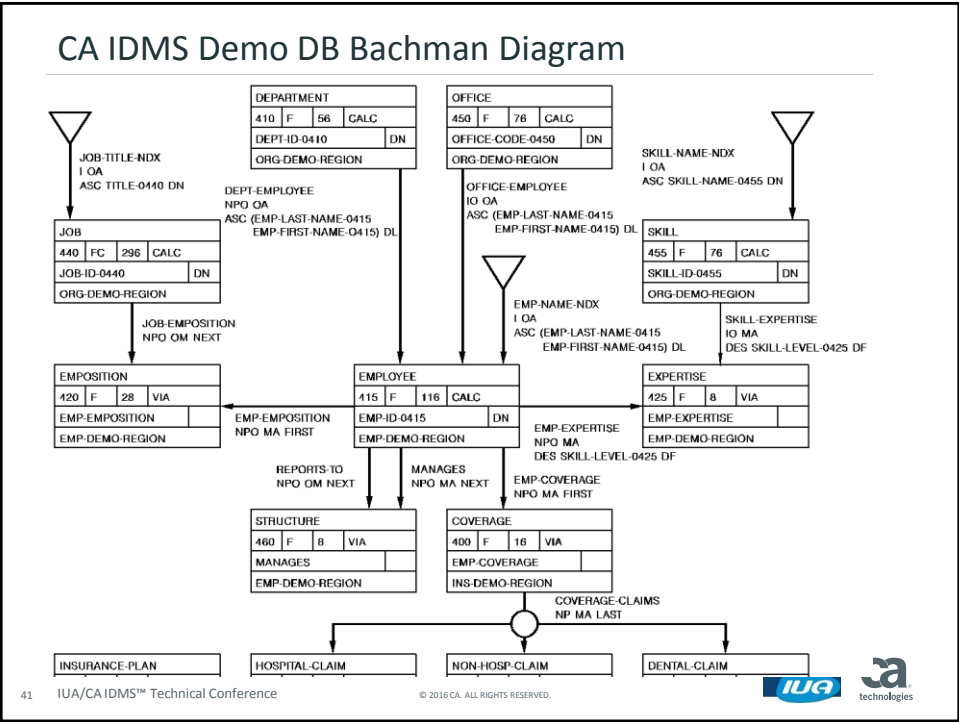
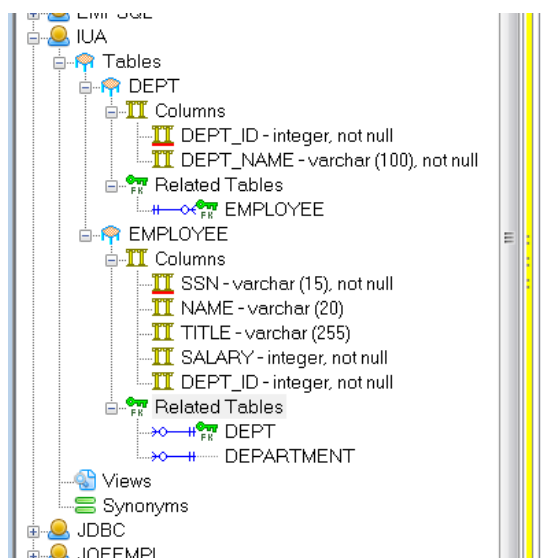


Table information



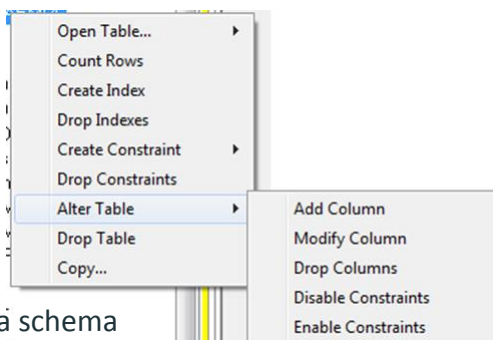
43 IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.



Table Manipulation

- Add Columns and Rows
- Edit Column Attributes
- Edit individual data
- Delete columns
- Create DDL to reproduce a schema

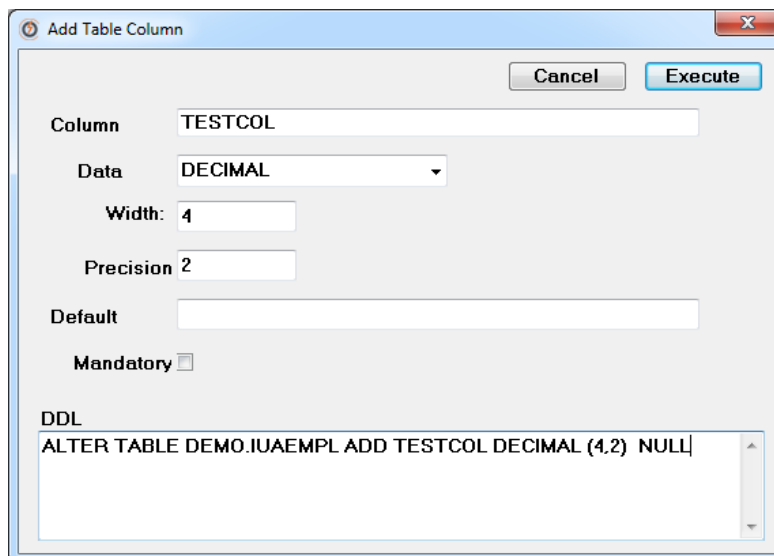


44 IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.



Create new column



Add Table Column

Column: TESTCOL

Data: DECIMAL

Width: 4

Precision: 2

Default:

Mandatory: ☐

DDL:
ALTER TABLE DEMO.IUAEMPL ADD TESTCOL DECIMAL (4,2) NULL

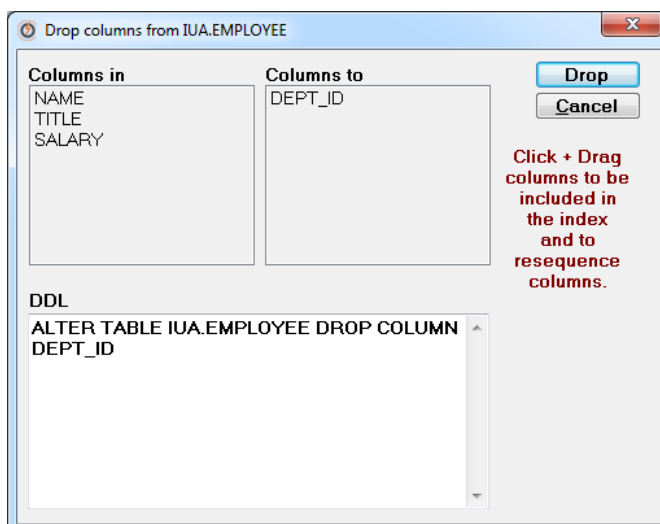
Buttons: Cancel, Execute

45 IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.



Drop Column



Drop columns from IUA.EMPLOYEE

Columns in	Columns to
NAME TITLE SALARY	DEPT_ID

Buttons: Drop, Cancel

DDL:
ALTER TABLE IUA.EMPLOYEE DROP COLUMN DEPT_ID

Click + Drag columns to be included in the index and to resequence columns.

46 IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.

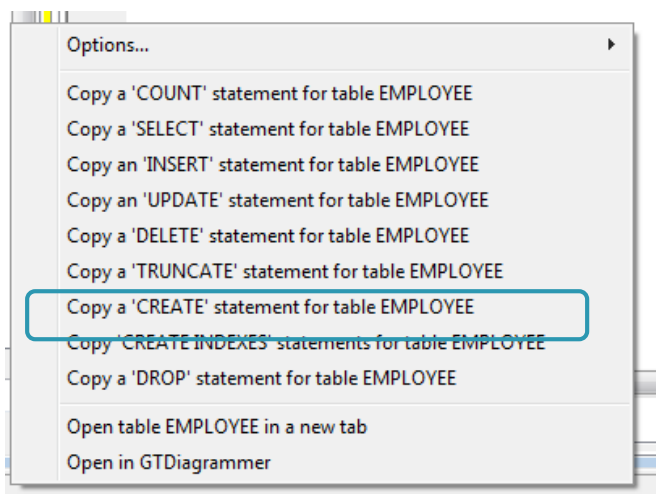


Transferring a table

- Create a duplicate table on another CV.
- Export the data from our source table to intermediate database



Copy table to new SYSQAS2



Copy Table Syntax

```

SQL #1 New
SQL Results Status
SELECT * FROM

CREATE TABLE iua.employee (
    ssn          varchar (15)    NOT NULL,
    name         varchar (20)    ,
    title        varchar (255)   ,
    salary       integer         NOT NULL,
    dept_id      integer         NOT NULL );

ALTER TABLE iua.employee
ADD CONSTRAINT genidx01112001 PRIMARY KEY (
    ssn
);

CREATE INDEX iua.genidx01112002 ON iua.employee (
    dept_id
);

```

49 IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.



Write data to Test Repository

edjo05 (Project: IDMSDemo)

SQL #1	SQL #2 [10]	New
SQL	Data in EMPLOYEE	Status

All 10 rows returned

Ssn	Name	Title	Salary	Dept Id
333445555	Mary	Product Owner	85000	1
444556666	Thomas	Sustaining	1000000	1
555667777	Christopher	Developer	66000	1
222334444	John	Developer	100000	1
111223333	James	Scrum Master	1600000	1
888990000	Mark	Calling	62000	2
777889999	Paul	Account Rep	250000	2
999001111	George	Account Rep	55000	2
666778888	Susan	Executive Account	99000	2
100112222	Lisa	Sales Engineer	35000	2

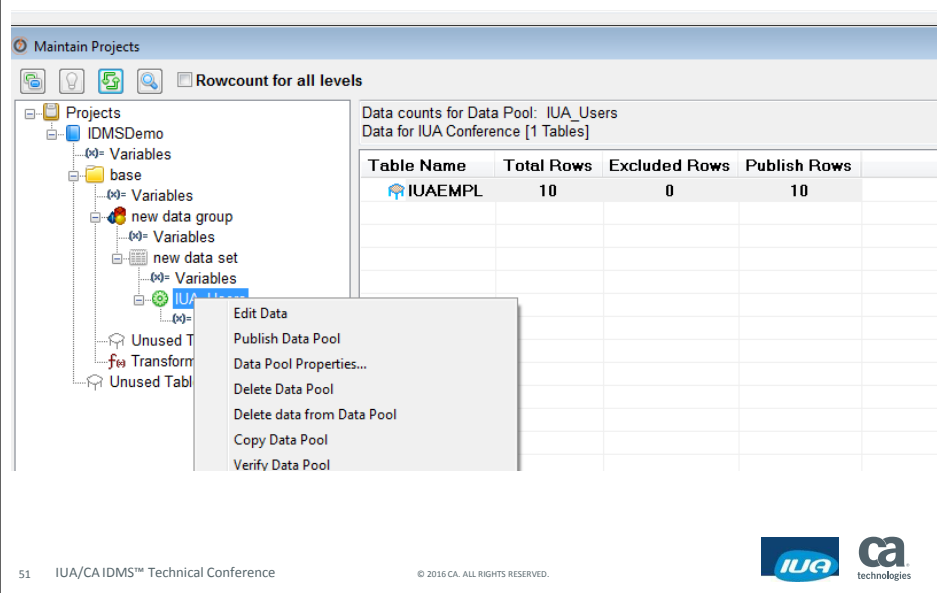
Show Context
 Show Context
 Change Context
 Write to Test Data Repository
 Write all to Test Data Repository
 Write all to Data Source
 Write all to Data Target

50 IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.



Edit data before transfer to new system



Maintain Projects

Rowcount for all levels

Data counts for Data Pool: IUA_Users
Data for IUA Conference [1 Tables]

Table Name	Total Rows	Excluded Rows	Publish Rows
IUAEMPL	10	0	10

Context Menu Options:

- Edit Data
- Publish Data Pool
- Data Pool Properties...
- Delete Data Pool
- Delete data from Data Pool
- Copy Data Pool
- Verify Data Pool

51 IUA/CA IDMS™ Technical Conference © 2016 CA. ALL RIGHTS RESERVED.

Original data

All 10 rows returned

Ssn	Name	Title	Salary	Dept I
333445555	Mary	Product Owner	85000	
444556666	Thomas	Sustaining	1000000	
555667777	Christopher	Developer	66000	
222334444	John	Developer	100000	
111223333	James	Scrum Master	1600000	
888990000	Mark	Calling	62000	
777889999	Paul	Account Rep	250000	
999001111	George	Account Rep	55000	
666778888	Susan	Executive Account	99000	
100112222	Lisa	Sales Engineer	35000	

- What do we want to change?

Data editing

- Insert data in bulk using various methods:
 - Seed values
 - Incremental
 - Random
- Data masking
 - Modify your data to hide sensitive information

53

IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.



Different ways of generating data

```

f Functions by type
+ f Numeric
+ f String
+ f Date and Time
+ f Code
+ f SQL
+ f Aggregate
+ f Control
+ f List of Values
f All Functions
  f 11proof(sequence,sign)
  f abs(number)
  f add(number,number)
  f addchecksum(number,method)
  f adddays(date,days)
  f addluhn(number)
  f addmicrosecs(timestamp,microseconds)
  
```

54

Seed values

Select Required Seed List

Seed Data Type	Cols	Rows
2 Digit Alpha	1	676
Australian Postal Codes	9	16742
BIC Codes	4	4713
BIC Codes	5	4713
Bank Transaction Types	2	22
Belgium Cities	1	50
Business Type	1	14
Canadian Cities	2	99
Canadian Postal-Codes	4	9434
Car Parts	2	1362
Companies	1	454
ComputerGames	1	3602
Country	1	263
Country Codes	4	239
Credit Card	1	2952
CreditCardType	1	3

Using seed values

Seed Data columnLinks To	Available column...
First Name	NAME	SSN
Middle Name		NAME
Title		TITLE
Sex		SALARY
		DEPT_ID
		Not linked

Using random values

IDMSDemo base Data Group: new data group Data Set: new data set

EMPLOYEE [10]

Data in IDMSDemo base Data Group: new data group Data Set: new data set Data Pool: IUA_Users

All 10 rows returned

Excl	Row	Ssn	Name	Title	Salary	Dept Id
<input type="checkbox"/>	1	@randdigits(1
<input type="checkbox"/>	2	@randdigits(1
<input type="checkbox"/>	3	@randdigits(1
<input type="checkbox"/>	4	@randdigits(1
<input type="checkbox"/>	5	@randdigits(1
<input type="checkbox"/>	6	@randdigits(2
<input type="checkbox"/>	7	@randdigits(2
<input type="checkbox"/>	8	@randdigits(2
<input type="checkbox"/>	9	@randdigits(2
<input type="checkbox"/>	10	@randdigits(2

Enter Min and Max for Random values

Column: SALARY

required data type: int

Min: 35000

Max: 1600000

OK

Cancel

Data in the test repository

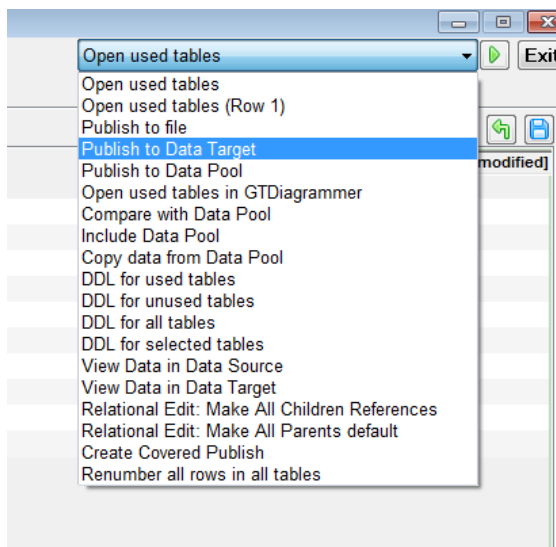
EMPLOYEE [10]

Data in IDMSDemo base Data Group: new data group Data Set: new data set Data Pool: IUA_Users

All 10 rows returned

Excl	Row	Ssn	Name	Title	Salary	Dept Id
<input type="checkbox"/>	1	@randdigits(9, Vasanti		Product Owner	436	1
<input type="checkbox"/>	2	@randdigits(9, Abhyudaya		Sustaining	2399	1
<input type="checkbox"/>	3	@randdigits(9, RanaId		Developer	931	1
<input type="checkbox"/>	4	@randdigits(9, Shraddha		Developer	1139	1
<input type="checkbox"/>	5	@randdigits(9, Kunto		Scrum Master	2328	1
<input type="checkbox"/>	6	@randdigits(9, Mutitu		Calling	836	2
<input type="checkbox"/>	7	@randdigits(9, Gangika		Account Rep	1718	2
<input type="checkbox"/>	8	@randdigits(9, Donnie		Account Rep	521	2
<input type="checkbox"/>	9	@randdigits(9, Ashling		Executive Account	842	2
<input type="checkbox"/>	10	@randdigits(9, Maia		Sales Engineer	1381	2

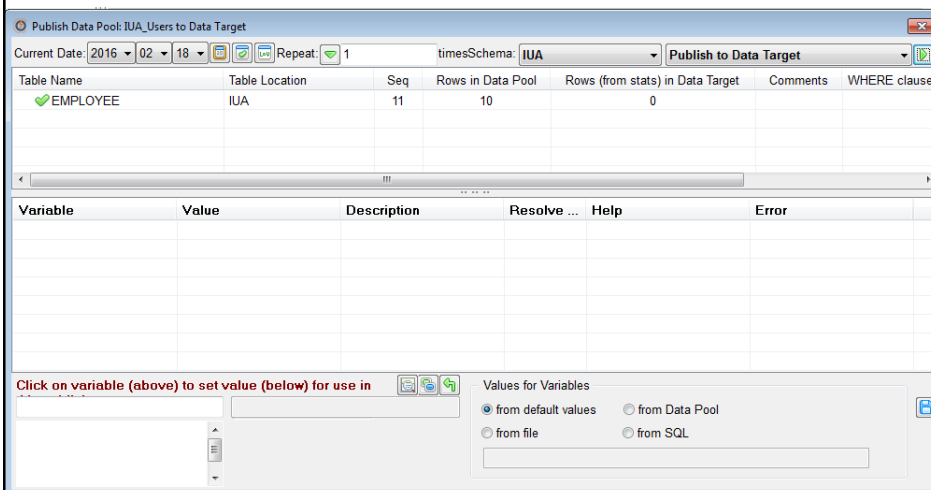
Publish to Target System



59 IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.

Add data to the Target (SYSQAS2)



60 IUA/CA IDMS™ Technical Conference

© 2016 CA. ALL RIGHTS RESERVED.

CA TDM Summary

- Visualized a network database
- Modified Columns
- Copied tables to a new system
- Manipulated data
- Populated table with new information

FOR INFORMATION PURPOSES ONLY Terms of this Presentation

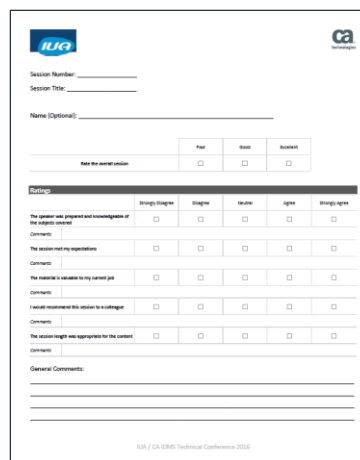
This presentation was based on current information and resource allocations as of May 2016 and is subject to change or withdrawal by CA at any time without notice. Notwithstanding anything in this presentation to the contrary, this presentation shall not serve to (i) affect the rights and/or obligations of CA or its licensees under any existing or future written license agreement or services agreement relating to any CA software product; or (ii) amend any product documentation or specifications for any CA software product. The development, release and timing of any features or functionality described in this presentation remain at CA's sole discretion. Notwithstanding anything in this presentation to the contrary, upon the general availability of any future CA product release referenced in this presentation, CA will make such release available (i) for sale to new licensees of such product; and (ii) to existing licensees of such product on a when and if-available basis as part of CA maintenance and support, and in the form of a regularly scheduled major product release. Such releases may be made available to current licensees of such product who are current subscribers to CA maintenance and support on a when and if-available basis. In the event of a conflict between the terms of this paragraph and any other information contained in this presentation, the terms of this paragraph shall govern.

Certain information in this presentation may outline CA's general product direction. All information in this presentation is for your informational purposes only and may not be incorporated into any contract. CA assumes no responsibility for the accuracy or completeness of the information. To the extent permitted by applicable law, CA provides this presentation "as is" without warranty of any kind, including without limitation, any implied warranties or merchantability, fitness for a particular purpose, or non-infringement. In no event will CA be liable for any loss or damage, direct or indirect, from the use of this document, including, without limitation, lost profits, lost investment, business interruption, goodwill, or lost data, even if CA is expressly advised in advance of the possibility of such damages. CA confidential and proprietary. No unauthorized copying or distribution permitted.

Questions and Answers

Please Complete a Session Evaluation Form

- The number for this session is **A06**
- After completing your session evaluation form, place it in the envelope at the front of the room



The form includes fields for Session Number, Session Title, and Name (Optional). It also has a section for Rate the session/lecture with checkboxes for Poor, Good, and Excellent. The main section is titled 'Rating' and contains five rows of evaluation questions, each with checkboxes for Strongly Dislike, Dislike, Neutral, Like, and Strongly Like. The questions are:

- The speaker was prepared and knowledgeable of the subject covered.
- The speaker met my expectations.
- The material is relevant to my current job.
- I would recommend this session to colleagues.
- The session length was appropriate for the content.

Each row has a 'Comments' field. At the bottom, there is a 'General Comments' section with a text area. The footer of the form reads 'IUA / CA IDMS Technical Conference 2016'.