CLARITY PPM
ACADEMY
2019

opemari

NSQL, still valuable even in the Modern UX

(Your query will run against this database)

NSQL

```
SELECT  @SELECT:DIM:USER_DEF:IMPLIED:FINANCIALS:e.entity:entity@,
            @SELECT:DIM_PROP:USER_DEF:IMPLIED:FINANCIALS:typ.name:prj_type@,
            @SELECT:DIM_PROP:USER_DEF:IMPLIED:FINANCIALS:NVL(tc.shortdesc, lu.name):trans_class@,
            @SELECT:DIM_PROP:USER_DEF:IMPLIED:FINANCIALS:to_char(w.transdate, 'yyyy'):rpt_year@,
            @SELECT:DIM_PROP:USER_DEF:IMPLIED:FINANCIALS:SUM(NVL(wv.totalcost,0)):amt@
            @SELECT:DIM_PROP:USER_DEF:IMPLIED:FINANCIALS:SUM(NVL(wv.totalcost,0)):amt@

FROM INV_INVESTMENTS inv
INNER JOIN ODF_CA_PROJECT p on inv.id = p.id
INNER JOIN CMN_LOOKUPS_V typ on P.obj_request_type = typ.lookup_code AND typ.lookup_type = 'OBJ_IDEA_PROJECT_TYPE'
LEFT OUTER JOIN CMN_LOOKUPS_V lu on lu.lookup_type = 'LOP_PORTLET_QUERY_TRANSLATIONS' and lu.language_code = 'en'
INNER JOIN ppa_wip w on inv.id = w.investment_id
INNER JOIN ppa_wip_values wv ON w.transno = wv.transno AND wv.currency_type = 'HOME'
LEFT OUTER JOIN transclass tc ON tc.transclass = w.transclass
INNER JOIN ENTITY e ON inv.entity_code = e.entity
INNER JOIN BIZ_COM_PERIODS b ON e.id = b.entity_id AND w.transdate BETWEEN b.start_date AND (b.end_date - 1) and b.
WHERE @FILTER@
AND @WHERE:SECURITY:PROJECT:inv.id@
GROUP BY e.entity, lu.name, typ.name, tc.shortdesc, to_char(w.transdate, 'yyyy')
```

Pemari Consulting Ltd

Andrew Litton

2007 - TODAY

About Me

I have been working in the Project Management Software consulting business since 1993, starting with Artemis, and then onto ABT, Niku and through it's rebranding to today and Clarity PPM.

Session Agenda

NSQL, still valuable even in the Modern UX

- Background
- Column Annotations
- Filters/Parameters
- Other Annotations

Background

NSQL, still valuable even in the Modern UX

Where is NSQL used?

- In lookups
  - when selecting Dynamic Query on lookup create
- Portlets – as a Query type data provider
  - One query can be used by many portlets
- XOG Query Soap Calls
  - Can call the query directly

Why is NSQL still useful with the Modern UX

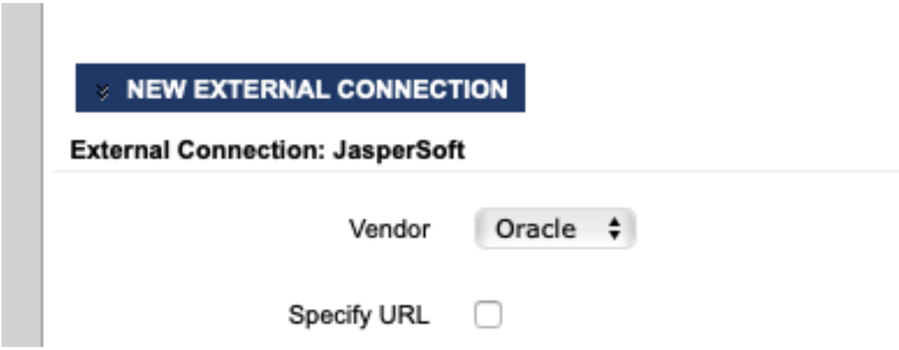
- Lookup attributes are available within the Modern UX

- Project Channels enable the embedding of Portlet Pages within the Project screens

Queries

- Queries created in Classic PPM within Administration Queries

- Queries can retrieve data from:
  - the Clarity PPM transactional database (Niku)
  - the Clarity PPM data warehouse (Data Warehouse)
  - Any other CSA configured database, i.e. Jaspersoft, using New External Connection

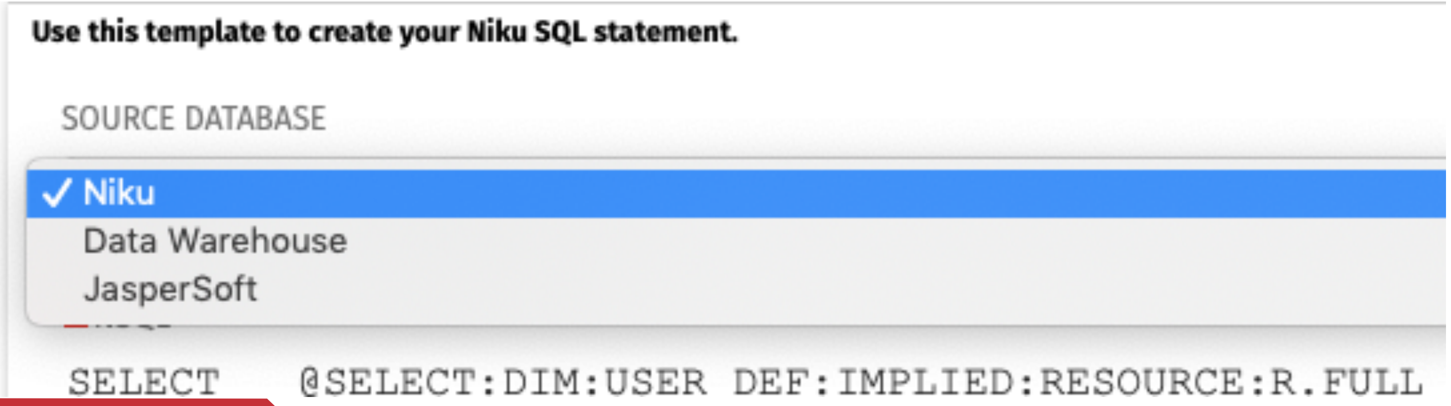


NEW EXTERNAL CONNECTION

External Connection: JasperSoft

Vendor: Oracle

Specify URL □



Use this template to create your Niku SQL statement.

SOURCE DATABASE

✓ Niku
Data Warehouse
JasperSoft

```
SELECT    @SELECT:DIM:USER DEF:IMPLIED:RESOURCE:R.FULL
```

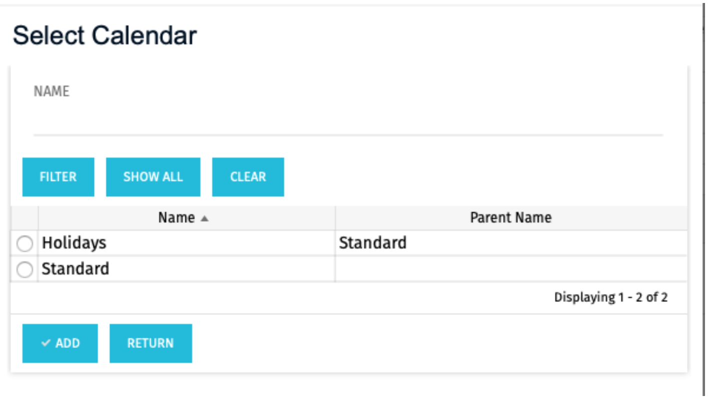
What is NSQL?

An annotated SQL Select statement

```
SELECT
  c.prid id,
  c.prname name,
  p.prname parentname,
  c.prmodtime last_updated_date
FROM prcalendar c
  LEFT OUTER JOIN prcalendar p
    ON c.prbasecalendarid = p.prid
WHERE c.prresourceid IS NULL
  AND 1=1
```

```
SELECT
  @SELECT:c.prid:id@,
  @SELECT:c.prname:name@,
  @SELECT:p.prname:parentname@,
  @SELECT:c.prmodtime:last_updated_date@
FROM  prcalendar c
  LEFT OUTER JOIN prcalendar p
    ON c.prbasecalendarid = p.prid
WHERE c.prresourceid IS NULL
  AND @FILTER@
```

Note annotations start and end with @



Select Calendar

NAME

	Name ▲	Parent Name
○	Holidays	Standard
○	Standard	

Displaying 1 - 2 of 2

Column annotations

NSQL, still valuable even in the Modern UX

Anatomy of a Select Statement

SELECT <columns>	Each column must use @SELECT:...@
FROM <table references>	Sub queries are ok to use
WHERE <where clauses>	Must contain at least @FILTER@
GROUP BY <group columns>	Aggregates data
HAVING <having clauses>	Filtering data uses @HAVING_FILTER@
ORDER BY <order columns>	Controllable in Clarity

SELECT Column Annotations

- 2 formats
  - For Lookups

    @SELECT:<Table.Field>:<Label>@
  - For Queries

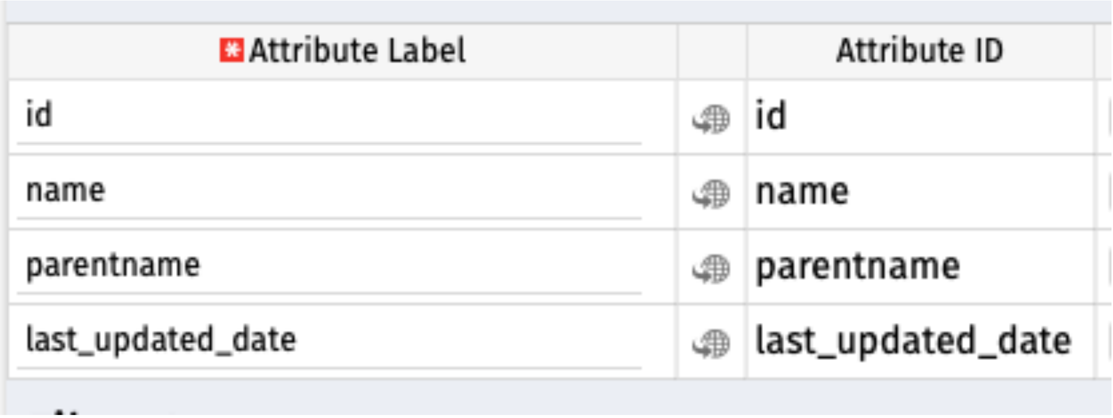
    @SELECT:DIM:USER_DEF:IMPLIED:<DIMENSION>:<Table.Field>:<Label>@

SELECT Column Annotations: Lookups

@SELECT:**<Table.Field>**:**<Label>**@

Table.Field	the SQL column
Label	used in Clarity as the Attribute ID and Label

```
SELECT
  @SELECT:c.prid:id@,
  @SELECT:c.prname:name@,
  @SELECT:p.prname:parentname@,
  @SELECT:c.prmodtime:last_updated_date@
FROM  prcalendar c
  LEFT OUTER JOIN prcalendar p
    ON c.prbasecalendarid = p.prid
WHERE c.prresourceid IS NULL
  AND @FILTER@
```



Attribute Label	Attribute ID
id	id
name	name
parentname	parentname
last_updated_date	last_updated_date

SELECT Column Annotations: Queries

- The format is a little more involved
- Annotation provides information to Clarity to enable grouping/pivoting/charting data in Portlets
- 3 types of Query SELECT annotation
  - Dimension
  - Dimension Property
  - Metric

SELECT Column Annotations: Dimension

@SELECT:**DIM**:USER_DEF:**IMPLIED**:**<DIMENSION>**:<Table.Field>:<Label>@

- **DIM** indicates the column is the primary key for the dimension
- There can be only one DIM column per dimension and there MUST be at least 1 dimension defined in an NSQL statement.
- **<DIMENSION>** is a user defined name for the dimension, i.e. Project or Resource
- **IMPLIED** tells Clarity to derive the attribute type from the SQL Result.

SELECT Column Annotations: Dimension Property

@SELECT:**DIM_PROP**:USER_DEF:IMPLIED:**<DIMENSION>**:<Table.Field>:<Label>@

- **DIM_PROP** indicates the column is a property of a dimension
- There can be many DIM_PROP columns per dimension
- **<DIMENSION>** is the name of the defined dimension, i.e. Project or Resource

SELECT Column Annotations: Metric

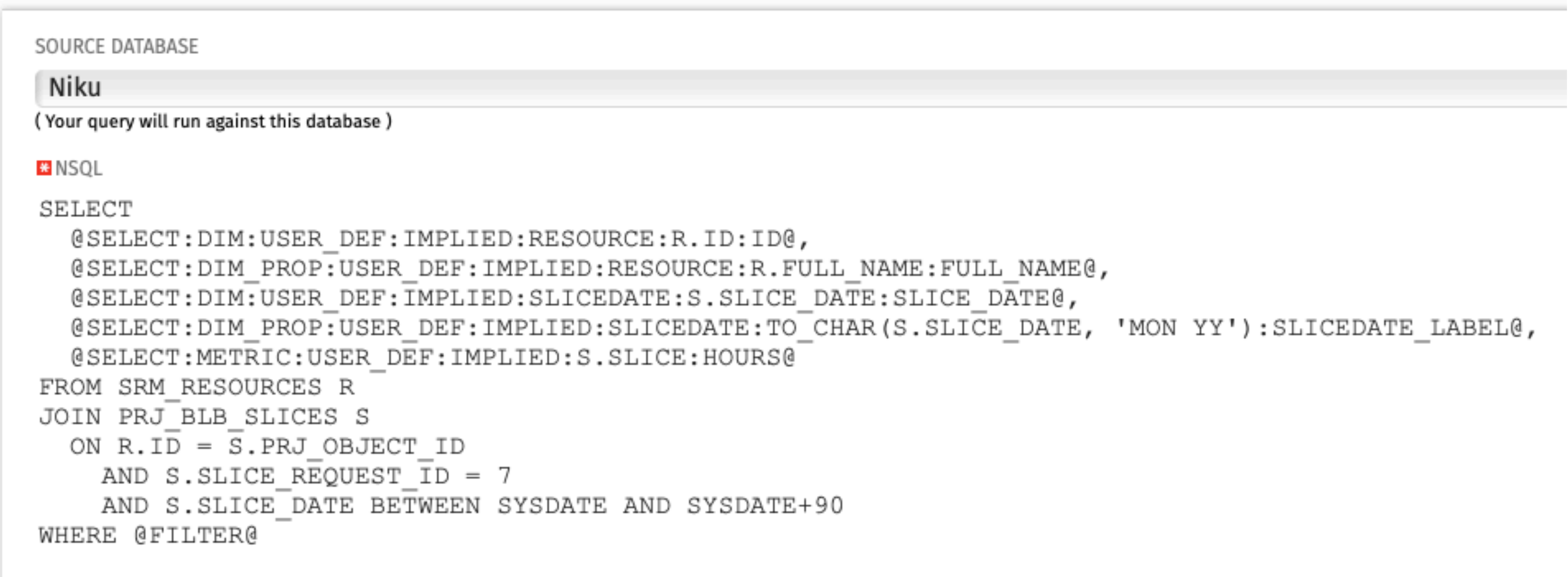
@SELECT:**METRIC**:USER_DEF:IMPLIED:<Table.Field>:<Label>**[:AGG]**@

- **METRIC** note no DIMENSION
- There can be many metric columns
- Can be totalled on a grid or displayed as a value on a chart
- **[:AGG]** optional construct and allows the metric to be filtered in the SQL HAVING clause

@SELECT:METRIC:USER_DEF:IMPLIED:COUNT(*):no_projects:AGG@

SELECT Column Annotations: Example Query

Resource Availability by Month



SOURCE DATABASE

Niku

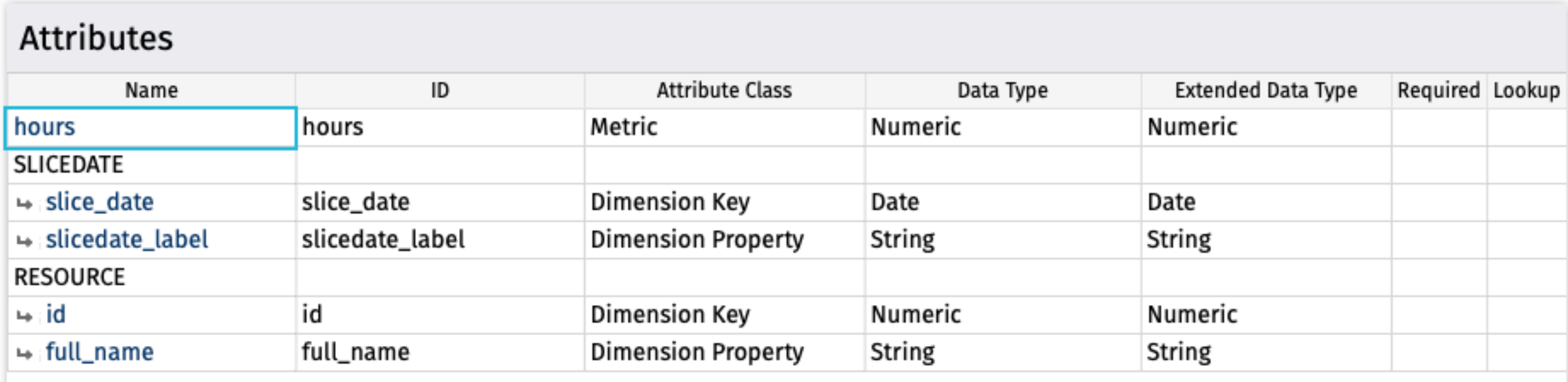
(Your query will run against this database)

NSQL

```
SELECT
  @SELECT:DIM:USER_DEF:IMPLIED:RESOURCE:R.ID:ID@,
  @SELECT:DIM_PROP:USER_DEF:IMPLIED:RESOURCE:R.FULL_NAME:FULL_NAME@,
  @SELECT:DIM:USER_DEF:IMPLIED:SLICEDATE:S.SLICE_DATE:SLICE_DATE@,
  @SELECT:DIM_PROP:USER_DEF:IMPLIED:SLICEDATE:TO_CHAR(S.SLICE_DATE, 'MON YY'):SLICEDATE_LABEL@,
  @SELECT:METRIC:USER_DEF:IMPLIED:S.SLICE:HOURS@
FROM SRM_RESOURCES R
JOIN PRJ_BLB_SLICES S
  ON R.ID = S.PRJ_OBJECT_ID
    AND S.SLICE_REQUEST_ID = 7
    AND S.SLICE_DATE BETWEEN SYSDATE AND SYSDATE+90
WHERE @FILTER@
```

SELECT Column Annotations: Example Query

Resource Availability by Month



Attributes

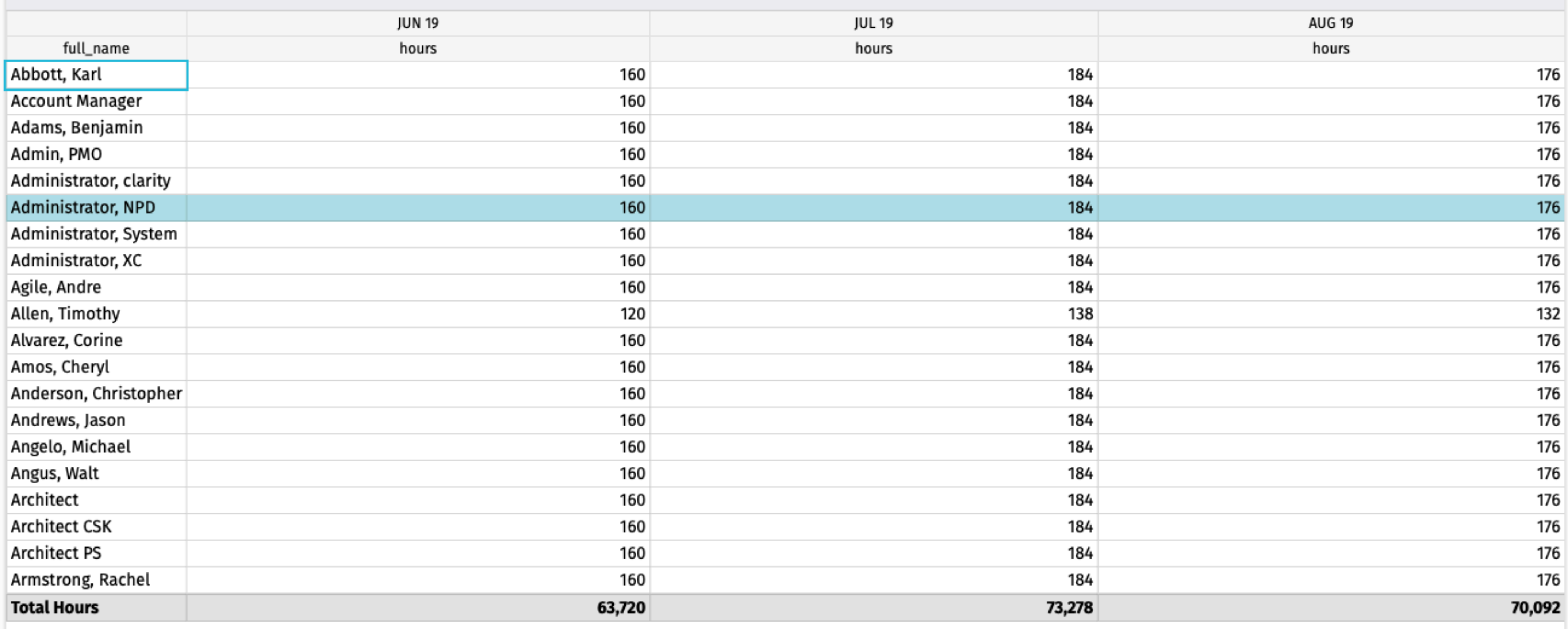
Name	ID	Attribute Class	Data Type	Extended Data Type	Required	Lookup
hours	hours	Metric	Numeric	Numeric		
SLICEDATE						
↳ slice_date	slice_date	Dimension Key	Date	Date		
↳ slicedate_label	slicedate_label	Dimension Property	String	String		
RESOURCE						
↳ id	id	Dimension Key	Numeric	Numeric		
↳ full_name	full_name	Dimension Property	String	String		

hours – is a metric

Two Dimensions – **SLICEDATE** and **RESOURCE**

SELECT Column Annotations: Example Grid Portlet

Resource Availability by Month



	JUN 19	JUL 19	AUG 19
full_name	hours	hours	hours
Abbott, Karl	160	184	176
Account Manager	160	184	176
Adams, Benjamin	160	184	176
Admin, PMO	160	184	176
Administrator, clarity	160	184	176
Administrator, NPD	160	184	176
Administrator, System	160	184	176
Administrator, XC	160	184	176
Agile, Andre	160	184	176
Allen, Timothy	120	138	132
Alvarez, Corine	160	184	176
Amos, Cheryl	160	184	176
Anderson, Christopher	160	184	176
Andrews, Jason	160	184	176
Angelo, Michael	160	184	176
Angus, Walt	160	184	176
Architect	160	184	176
Architect CSK	160	184	176
Architect PS	160	184	176
Armstrong, Rachel	160	184	176
Total Hours	**63,720**	**73,278**	**70,092**

Filters/Parameters

NSQL, still valuable even in the Modern UX

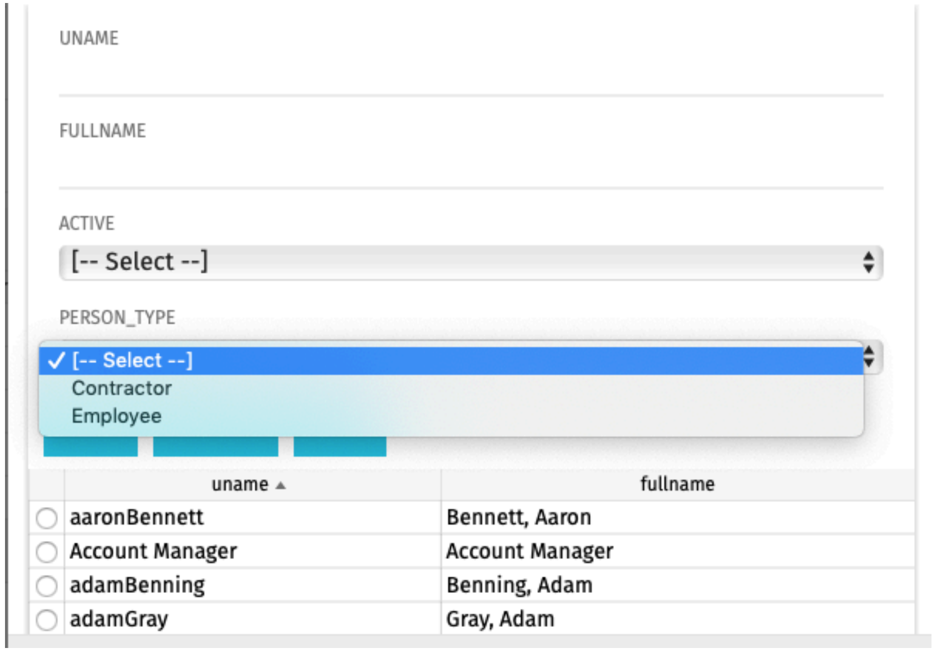
WHERE Annotations: FILTER

@FILTER@

- @FILTER@ allows Clarity to filter the columns defined in the @SELECT@ annotations

- @FILTER@ is replaced on execution with WHERE clauses based on the portlet or lookup filters

```
SELECT
  @SELECT:R.ID:ID@,
  @SELECT:R.UNIQUE_NAME:UNAME@,
  @SELECT:R.FULL_NAME:FULLNAME@,
  @SELECT:R.IS_ACTIVE:ACTIVE@,
  @SELECT:R.PERSON_TYPE:PERSON_TYPE@
FROM SRM_RESOURCES R
WHERE @FILTER@
```



UNAME

FULLNAME

ACTIVE

[-- Select --]

PERSON_TYPE

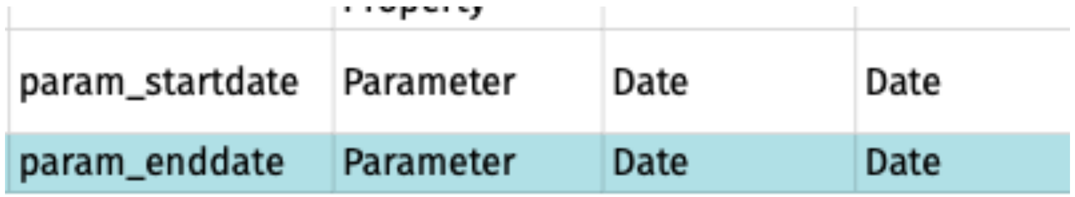
✓ [-- Select --]
Contractor
Employee

	uname ▲	fullname
○	aaronBennett	Bennett, Aaron
○	Account Manager	Account Manager
○	adamBenning	Benning, Adam
○	adamGray	Gray, Adam

WHERE Annotations: PARAMS

@WHERE:**PARAM**:…….@

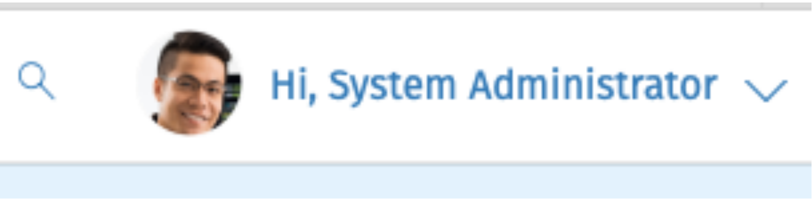
- Can be located in different sub queries
- Do not need to be one of the @SELECT@ annotations
- A number of sources
  - From filters
  - Built in
  - From the page URL



param_startdate	Parameter	Date	Date
param_enddate	Parameter	Date	Date

WHERE Annotations: Built In Parameters

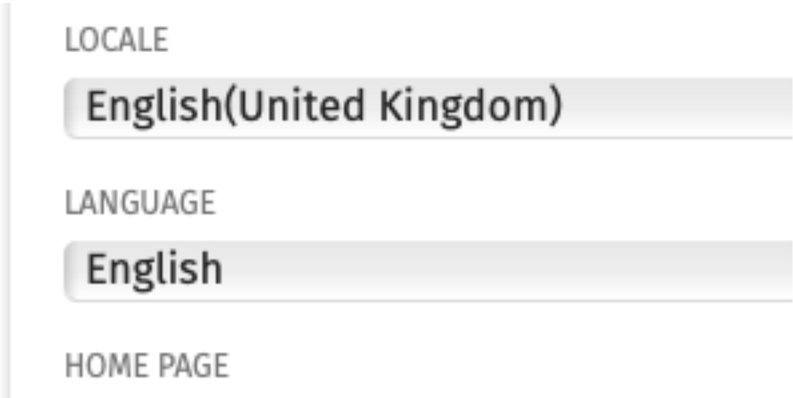
- Commonly used for security and the selection of the relevant lookup value language



Hi, System Administrator

@WHERE:PARAM**:USER_ID**

@WHERE :PARAM**:USER_NAME@**



LOCALE

English(United Kingdom)

LANGUAGE

English

HOME PAGE

@WHERE :PARAM**:LOCALE@**

@WHERE :PARAM**:LANGUAGE@**

WHERE Annotations: Built In Lookup Example

```
SELECT
 @SELECT:R.ID:ID@,
 @SELECT:R.UNIQUE_NAME:UNAME@,
 @SELECT:R.FULL_NAME:FULLNAME@,
 @SELECT:L.NAME:PERSON_TYPE@,
 @SELECT:R.PERSON_TYPE:PERSON_TYPE_ID@,
 @SELECT:L.LANGUAGE_CODE:LANGUAGE_CODE@
FROM SRM_RESOURCES R
 LEFT OUTER JOIN CMN_LOOKUPS_V L
  ON R.PERSON_TYPE=L.ID
   AND L.LOOKUP_TYPE='SRM_RESOURCE_TYPE'
   AND L.LANGUAGE_CODE=@WHERE:PARAM:LANGUAGE@
WHERE R.IS_ACTIVE=1
 AND @FILTER@
```

```
SELECT
 @SELECT:R.ID:ID@,
 @SELECT:R.UNIQUE_NAME:UNAME@,
 @SELECT:R.FULL_NAME:FULLNAME@,
 @SELECT:L.NAME:PERSON_TYPE@,
 @SELECT:R.PERSON_TYPE:PERSON_TYPE_ID@,
 @WHERE:PARAM:LANGUAGE@ LANGUAGE_CODE
FROM SRM_RESOURCES R
 LEFT OUTER JOIN CMN_LOOKUPS_V L
  ON R.PERSON_TYPE=L.ID
   AND L.LOOKUP_TYPE='SRM_RESOURCE_TYPE'
   AND L.LANGUAGE_CODE=@WHERE:PARAM:LANGUAGE@
WHERE R.IS_ACTIVE=1
 AND @FILTER@
```

WHERE Annotations: User Defined Parameters

@WHERE:**PARAM**:USER_DEF:**<DATA_TYPE>**:**PARAM_NAME@**

@WHERE:**PARAM**:USER_DEF:**<DATA_TYPE>**:**<COLUMN_NAME>**:**PARAM_NAME@**

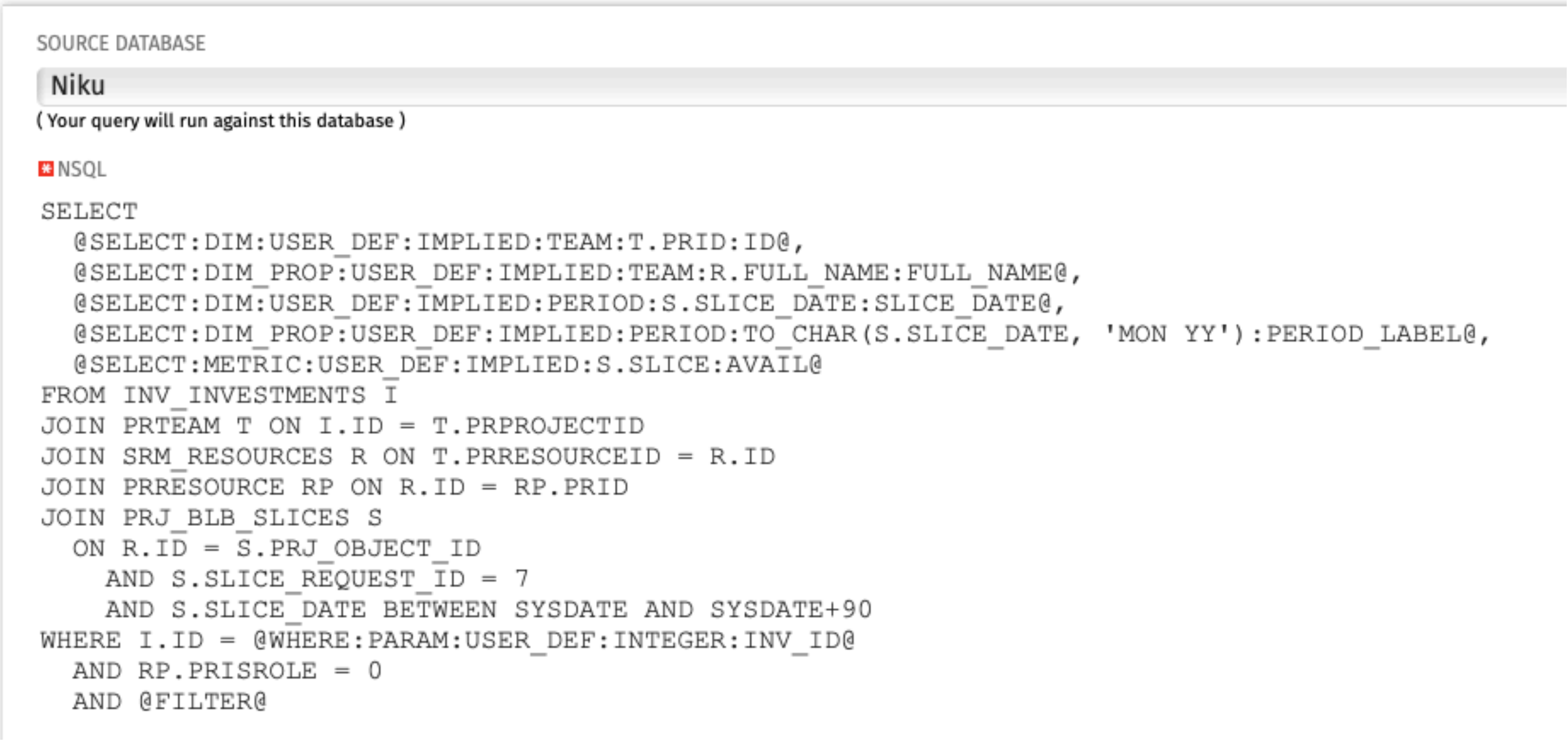
- **PARAM_NAME** is the name of the attribute displayed within Clarity PPM
- **<DATA_TYPE>** value can be **STRING**, **INTEGER**, or, **DATE**

  Although FLOAT is accepted Query displays this as a numeric

- 2nd Form optional and enables support for Multiple Selection.
  - **COLUMN_NAME** is the SQL column name for the condition

WHERE Annotations: User Defined Example 1

Investment Team Availability by Month



SOURCE DATABASE

Niku

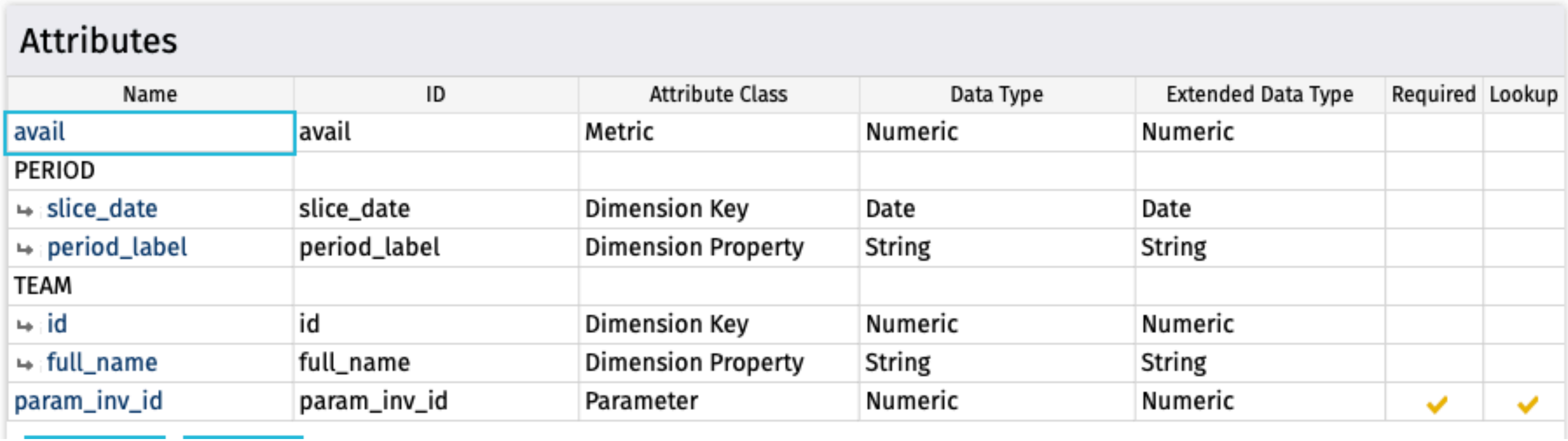
(Your query will run against this database)

NSQL

```
SELECT
  @SELECT:DIM:USER_DEF:IMPLIED:TEAM:T.PRID:ID@,
  @SELECT:DIM_PROP:USER_DEF:IMPLIED:TEAM:R.FULL_NAME:FULL_NAME@,
  @SELECT:DIM:USER_DEF:IMPLIED:PERIOD:S.SLICE_DATE:SLICE_DATE@,
  @SELECT:DIM_PROP:USER_DEF:IMPLIED:PERIOD:TO_CHAR(S.SLICE_DATE, 'MON YY'):PERIOD_LABEL@,
  @SELECT:METRIC:USER_DEF:IMPLIED:S.SLICE:AVAIL@
FROM INV_INVESTMENTS I
JOIN PRTEAM T ON I.ID = T.PRPROJECTID
JOIN SRM_RESOURCES R ON T.PRRESOURCEID = R.ID
JOIN PRRESOURCE RP ON R.ID = RP.PRID
JOIN PRJ_BLB_SLICES S
  ON R.ID = S.PRJ_OBJECT_ID
    AND S.SLICE_REQUEST_ID = 7
    AND S.SLICE_DATE BETWEEN SYSDATE AND SYSDATE+90
WHERE I.ID = @WHERE:PARAM:USER_DEF:INTEGER:INV_ID@
  AND RP.PRISROLE = 0
  AND @FILTER@
```

WHERE Annotations: User Defined Example 1

Investment Team Availability by Month

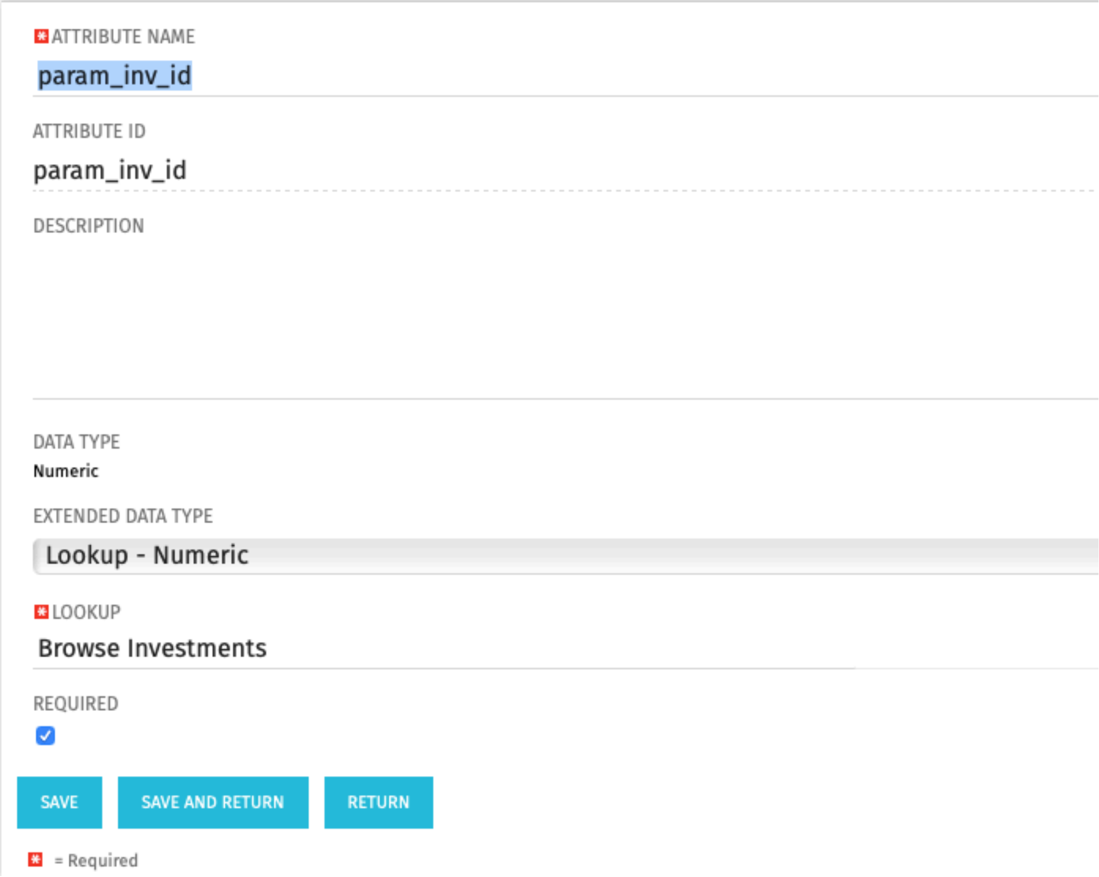


Attributes

Name	ID	Attribute Class	Data Type	Extended Data Type	Required	Lookup
avail	avail	Metric	Numeric	Numeric		
PERIOD						
↳ slice_date	slice_date	Dimension Key	Date	Date		
↳ period_label	period_label	Dimension Property	String	String		
TEAM						
↳ id	id	Dimension Key	Numeric	Numeric		
↳ full_name	full_name	Dimension Property	String	String		
param_inv_id	param_inv_id	Parameter	Numeric	Numeric	✓	✓

WHERE Annotations: User Defined Example 1

Investment Team Availability by Month



☒ATTRIBUTE NAME
param_inv_id

ATTRIBUTE ID
param_inv_id

DESCRIPTION

DATA TYPE
Numeric

EXTENDED DATA TYPE
Lookup - Numeric

☒LOOKUP
Browse Investments

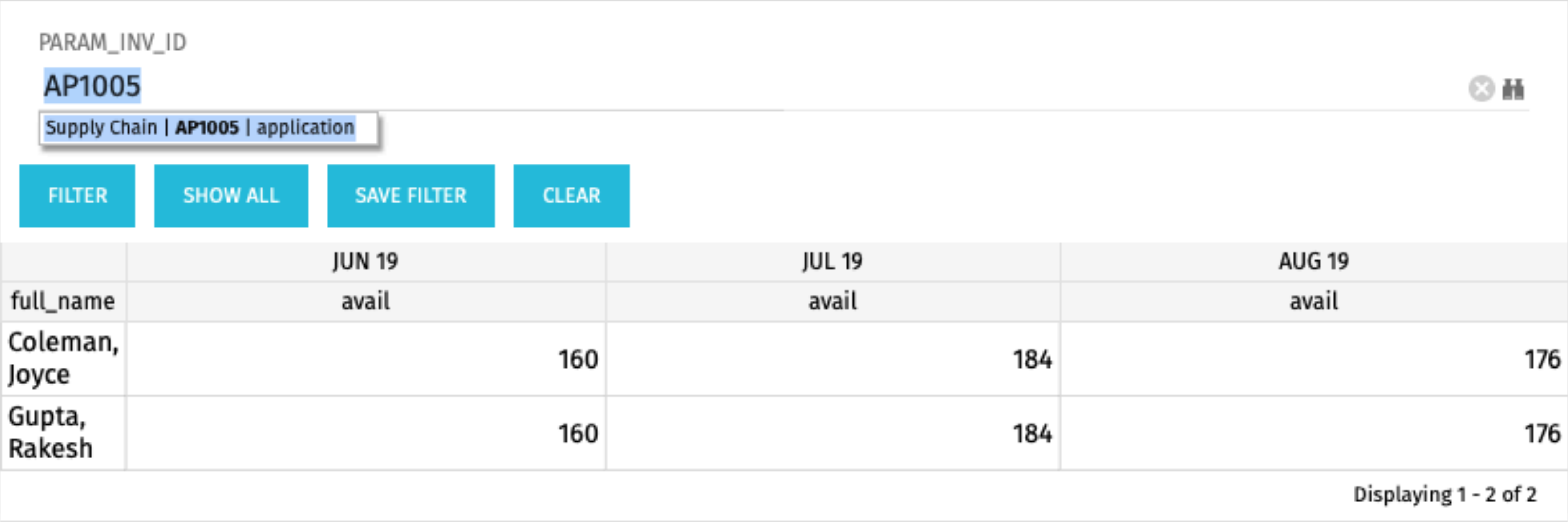
REQUIRED
☑

☒ = Required

- Change Field Extended type to Lookup - Numeric
- Pick the relevant lookup
- Portlet filter can display a browse field for the project selection

WHERE Annotations: User Defined Example 1

Investment Team Availability by Month



PARAM_INV_ID

AP1005

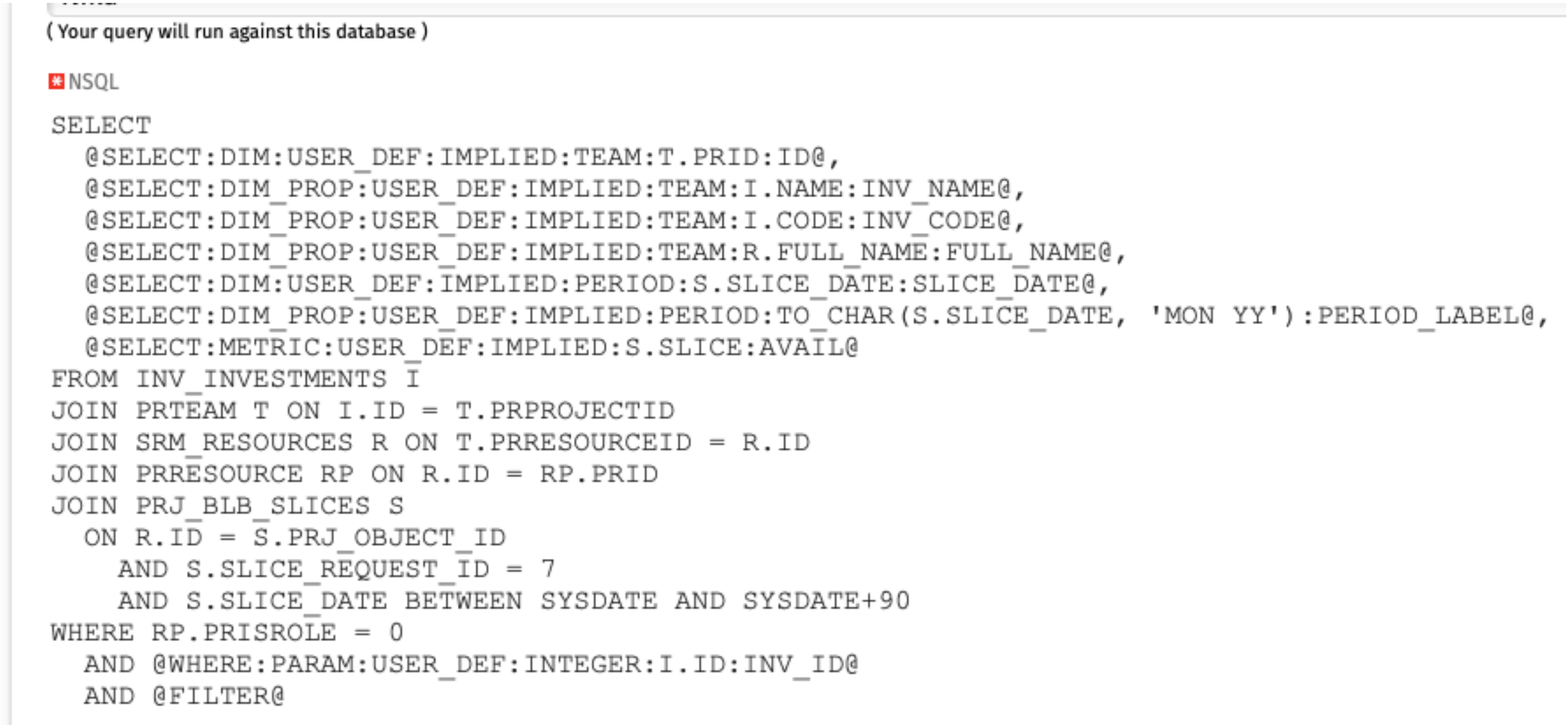
Supply Chain | **AP1005** | application

	JUN 19	JUL 19	AUG 19
full_name	avail	avail	avail
Coleman, Joyce	160	184	176
Gupta, Rakesh	160	184	176

Displaying 1 - 2 of 2

WHERE Annotations: User Defined Example 2

Investment Team Availability by Month



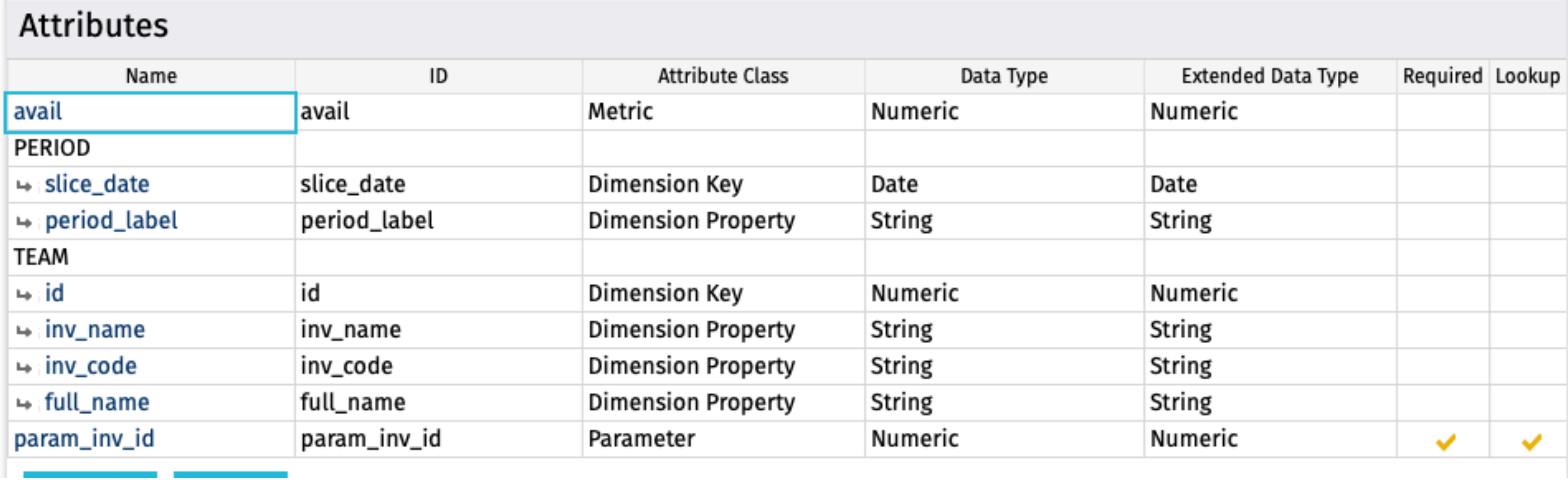
(Your query will run against this database)

✱NSQL

```
SELECT
  @SELECT:DIM:USER_DEF:IMPLIED:TEAM:T.PRID:ID@,
  @SELECT:DIM_PROP:USER_DEF:IMPLIED:TEAM:I.NAME:INV_NAME@,
  @SELECT:DIM_PROP:USER_DEF:IMPLIED:TEAM:I.CODE:INV_CODE@,
  @SELECT:DIM_PROP:USER_DEF:IMPLIED:TEAM:R.FULL_NAME:FULL_NAME@,
  @SELECT:DIM:USER_DEF:IMPLIED:PERIOD:S.SLICE_DATE:SLICE_DATE@,
  @SELECT:DIM_PROP:USER_DEF:IMPLIED:PERIOD:TO_CHAR(S.SLICE_DATE, 'MON YY'):PERIOD_LABEL@,
  @SELECT:METRIC:USER_DEF:IMPLIED:S.SLICE:AVAIL@
FROM INV_INVESTMENTS I
JOIN PRTEAM T ON I.ID = T.PRPROJECTID
JOIN SRM_RESOURCES R ON T.PRRESOURCEID = R.ID
JOIN PRRESOURCE RP ON R.ID = RP.PRID
JOIN PRJ_BLB_SLICES S
  ON R.ID = S.PRJ_OBJECT_ID
    AND S.SLICE_REQUEST_ID = 7
    AND S.SLICE_DATE BETWEEN SYSDATE AND SYSDATE+90
WHERE RP.PRISROLE = 0
  AND @WHERE:PARAM:USER_DEF:INTEGER:I.ID:INV_ID@
  AND @FILTER@
```

WHERE Annotations: User Defined Example 2

Investment Team Availability by Month

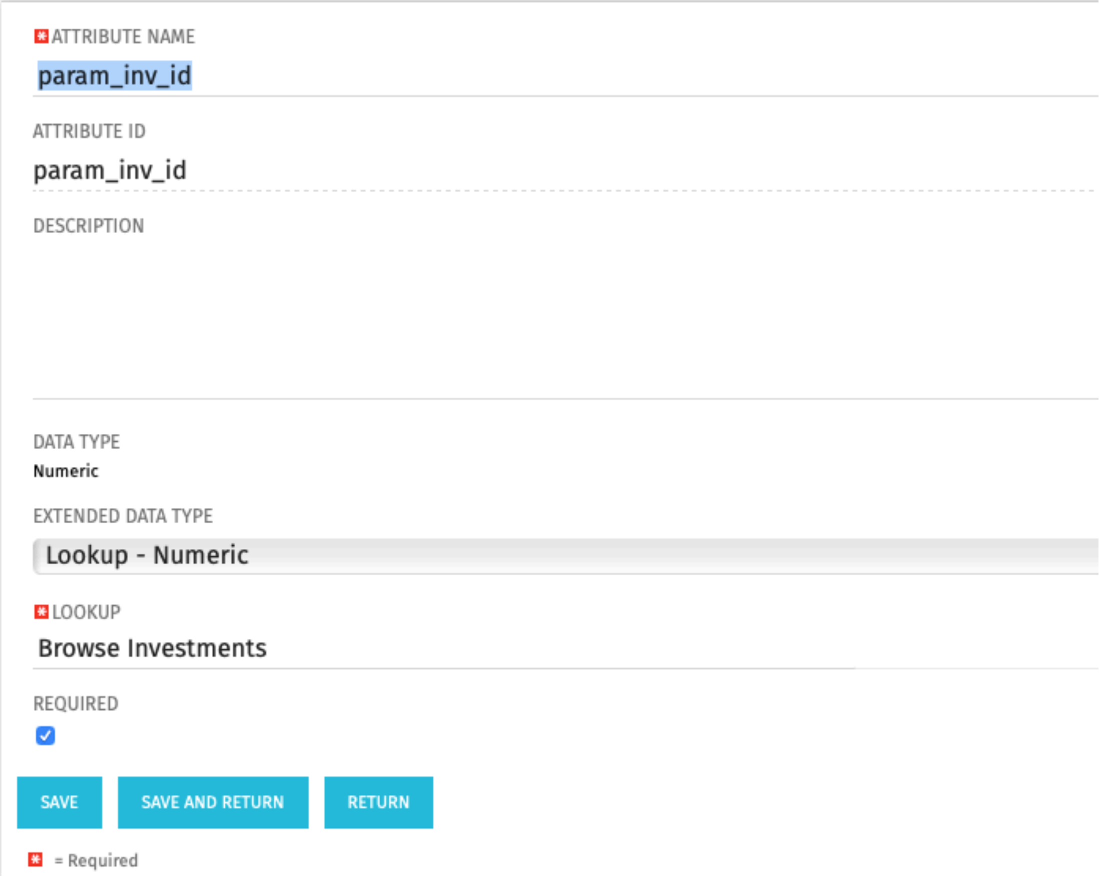


Attributes

Name	ID	Attribute Class	Data Type	Extended Data Type	Required	Lookup
avail	avail	Metric	Numeric	Numeric		
PERIOD						
↳ slice_date	slice_date	Dimension Key	Date	Date		
↳ period_label	period_label	Dimension Property	String	String		
TEAM						
↳ id	id	Dimension Key	Numeric	Numeric		
↳ inv_name	inv_name	Dimension Property	String	String		
↳ inv_code	inv_code	Dimension Property	String	String		
↳ full_name	full_name	Dimension Property	String	String		
param_inv_id	param_inv_id	Parameter	Numeric	Numeric	✓	✓

WHERE Annotations: User Defined Example 2

Investment Team Availability by Month



ATTRIBUTE NAME
param_inv_id

ATTRIBUTE ID
param_inv_id

DESCRIPTION

DATA TYPE
Numeric

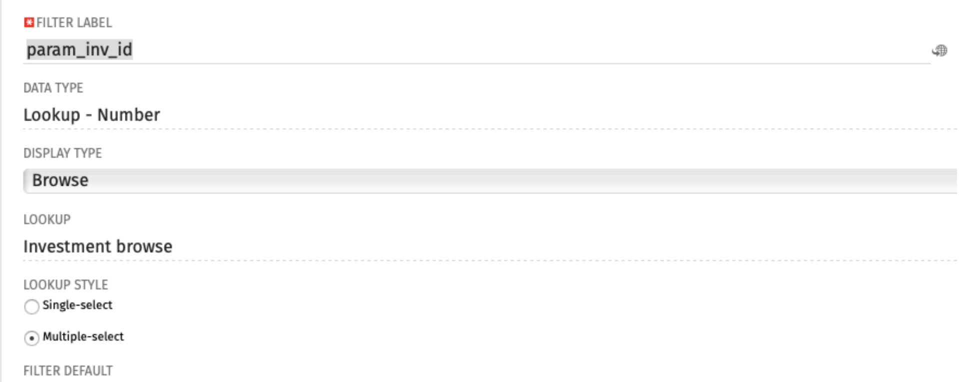
EXTENDED DATA TYPE
Lookup - Numeric

LOOKUP
Browse Investments

REQUIRED
☑

= Required

- Change Field Extended type to Lookup - Numeric
- Pick the relevant lookup
- Portlet filter can use a Multiple Select browse field for the project selection



FILTER LABEL
param_inv_id

DATA TYPE
Lookup - Number

DISPLAY TYPE
Browse

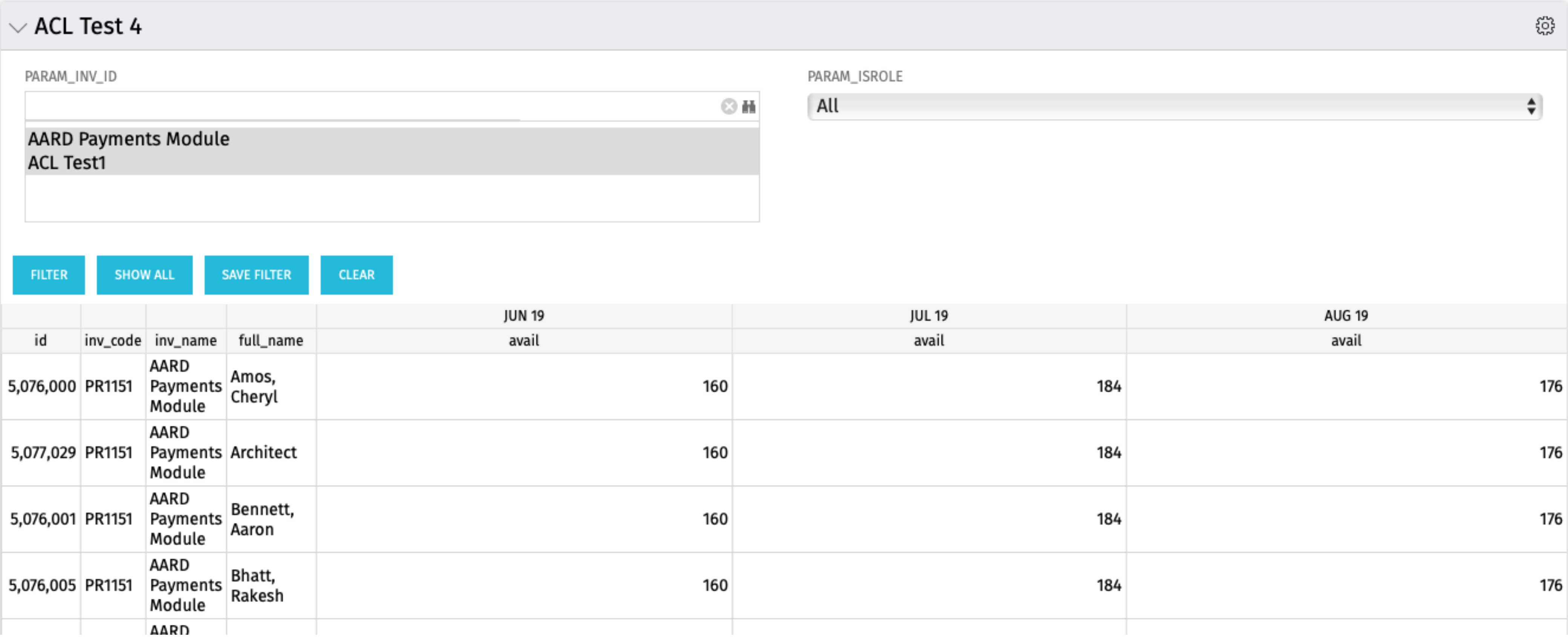
LOOKUP
Investment browse

LOOKUP STYLE
○ Single-select
● Multiple-select

FILTER DEFAULT

WHERE Annotations: User Defined Example 2

Investment Team Availability by Month



ACL Test 4

PARAM_INV_ID

AARD Payments Module
ACL Test1

PARAM_ISROLE: All

				JUN 19	JUL 19	AUG 19
id	inv_code	inv_name	full_name	avail	avail	avail
5,076,000	PR1151	AARD Payments Module	Amos, Cheryl	160	184	176
5,077,029	PR1151	AARD Payments Module	Architect	160	184	176
5,076,001	PR1151	AARD Payments Module	Bennett, Aaron	160	184	176
5,076,005	PR1151	AARD Payments Module	Bhatt, Rakesh	160	184	176
		AARD				

WHERE Annotations: XML Parameters

@WHERE:**PARAM**:XML:<DATA_TYPE>:<xpath>@

- Obtains value from the URL of the page of the portlet
- Example : A dashboard page within Project with a URL

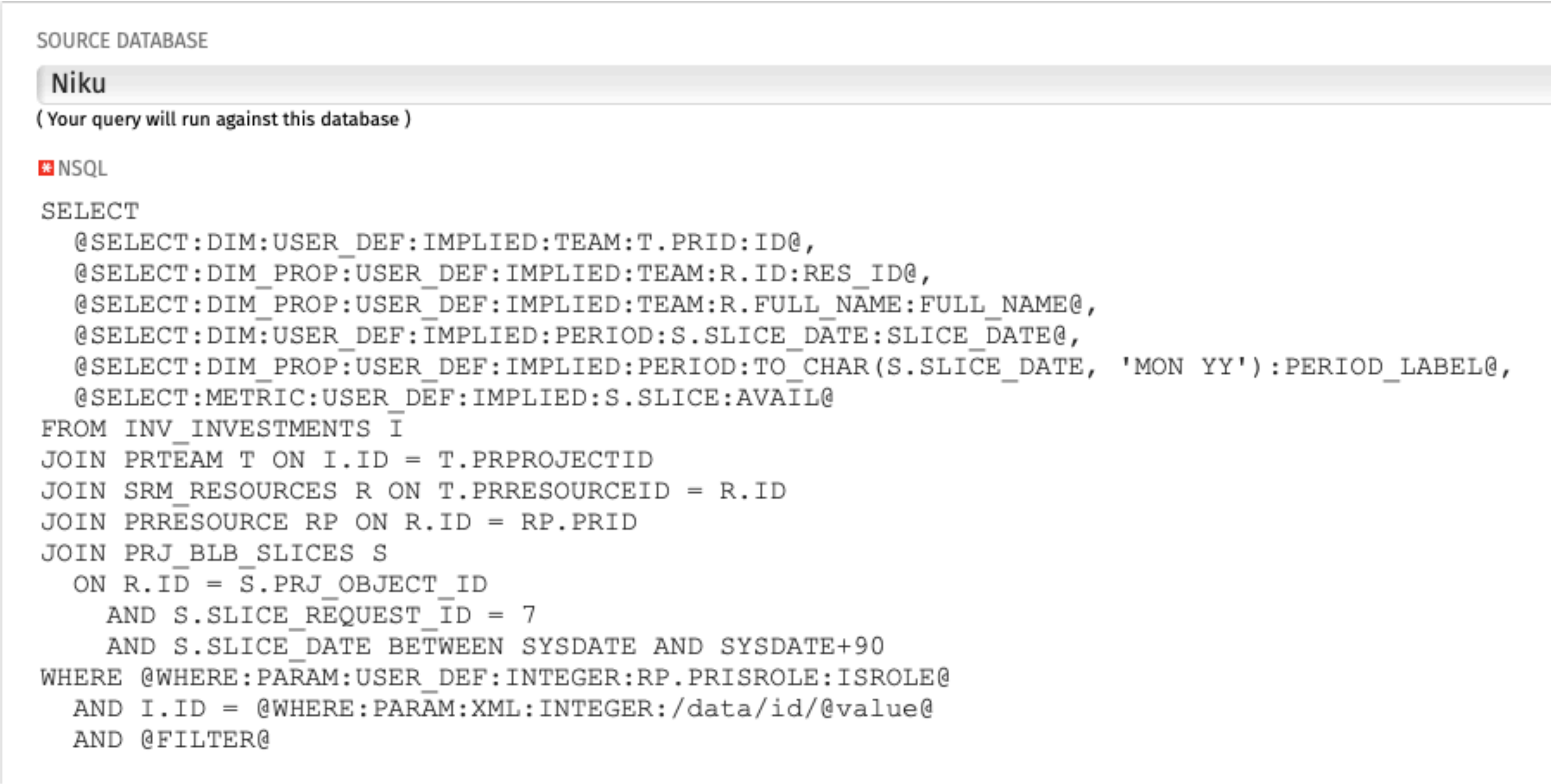
  https://centos.vm.pemari.com/niku/nu#action:SYSTEM5130026&id=5048013

- So define an XML Parameter to read the id and only display values for the project

  @WHERE:PARAM:XML:INTEGER:/data/id/@value@

WHERE Annotations: XML Example

Investment Team Availability by Month



SOURCE DATABASE

Niku

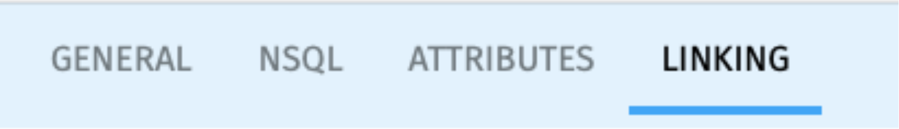
(Your query will run against this database)

NSQL

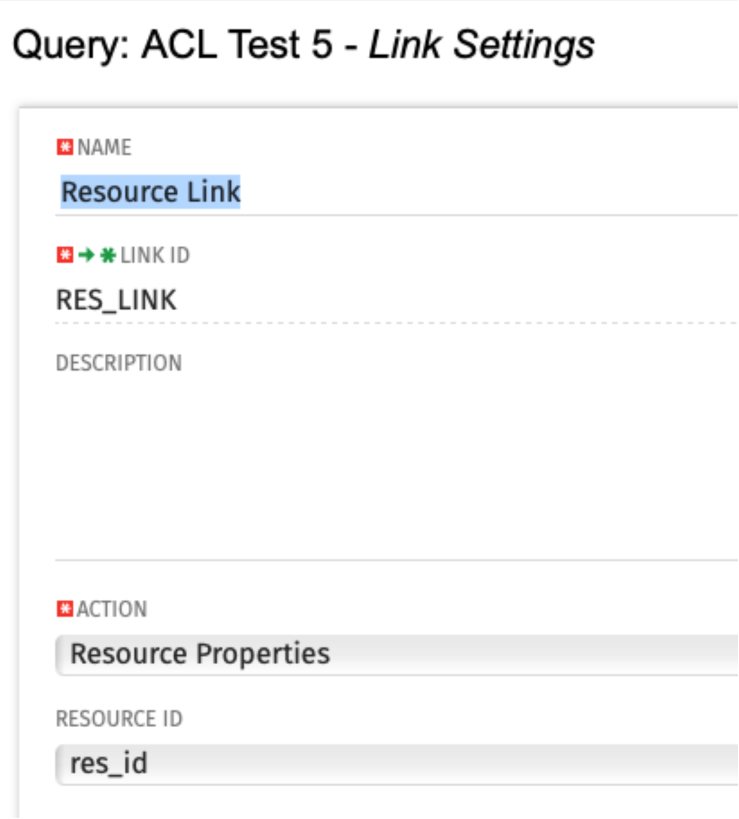
```
SELECT
  @SELECT:DIM:USER_DEF:IMPLIED:TEAM:T.PRID:ID@,
  @SELECT:DIM_PROP:USER_DEF:IMPLIED:TEAM:R.ID:RES_ID@,
  @SELECT:DIM_PROP:USER_DEF:IMPLIED:TEAM:R.FULL_NAME:FULL_NAME@,
  @SELECT:DIM:USER_DEF:IMPLIED:PERIOD:S.SLICE_DATE:SLICE_DATE@,
  @SELECT:DIM_PROP:USER_DEF:IMPLIED:PERIOD:TO_CHAR(S.SLICE_DATE, 'MON YY'):PERIOD_LABEL@,
  @SELECT:METRIC:USER_DEF:IMPLIED:S.SLICE:AVAIL@
FROM INV_INVESTMENTS I
JOIN PRTEAM T ON I.ID = T.PRPROJECTID
JOIN SRM_RESOURCES R ON T.PRRESOURCEID = R.ID
JOIN PRRESOURCE RP ON R.ID = RP.PRID
JOIN PRJ_BLB_SLICES S
  ON R.ID = S.PRJ_OBJECT_ID
    AND S.SLICE_REQUEST_ID = 7
    AND S.SLICE_DATE BETWEEN SYSDATE AND SYSDATE+90
WHERE @WHERE:PARAM:USER_DEF:INTEGER:RP.PRISROLE:ISROLE@
  AND I.ID = @WHERE:PARAM:XML:INTEGER:/data/id/@value@
  AND @FILTER@
```

WHERE Annotations: XML Example

Investment Team Availability by Month



Note: Queries can have links defined as well to enable linking from the portlet to another page



Query: ACL Test 5 - *Link Settings*

NAME
Resource Link

LINK ID
RES_LINK

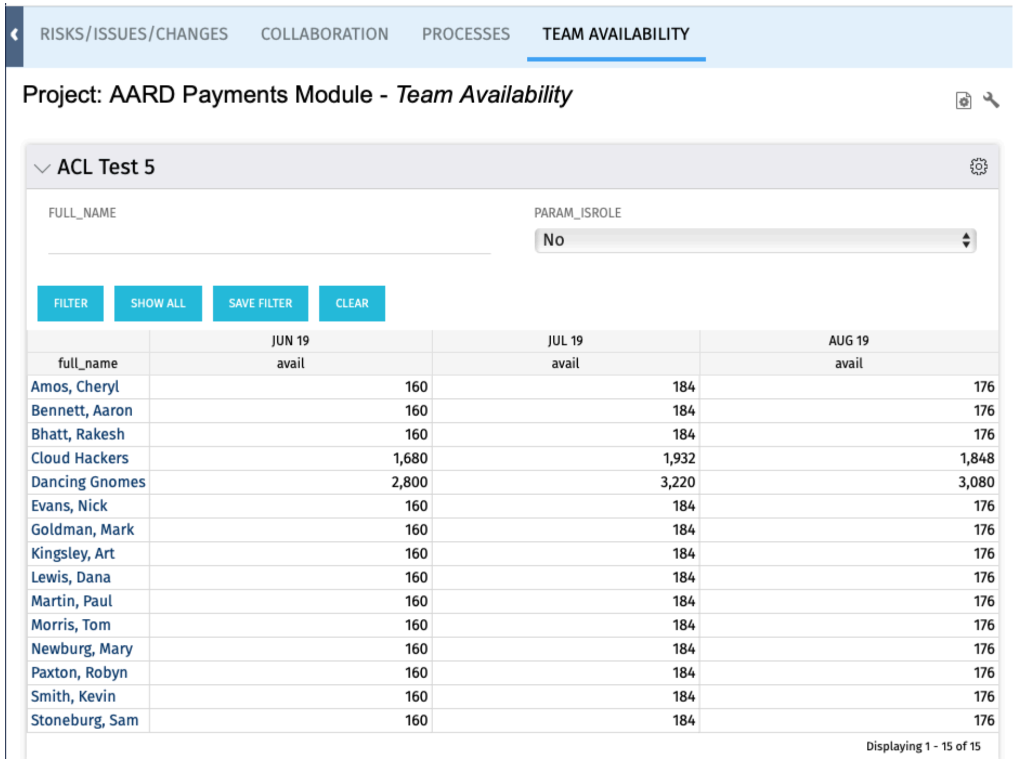
DESCRIPTION

ACTION
Resource Properties

RESOURCE ID
res_id

WHERE Annotations: XML Example

Investment Team Availability by Month



Project: AARD Payments Module - *Team Availability*

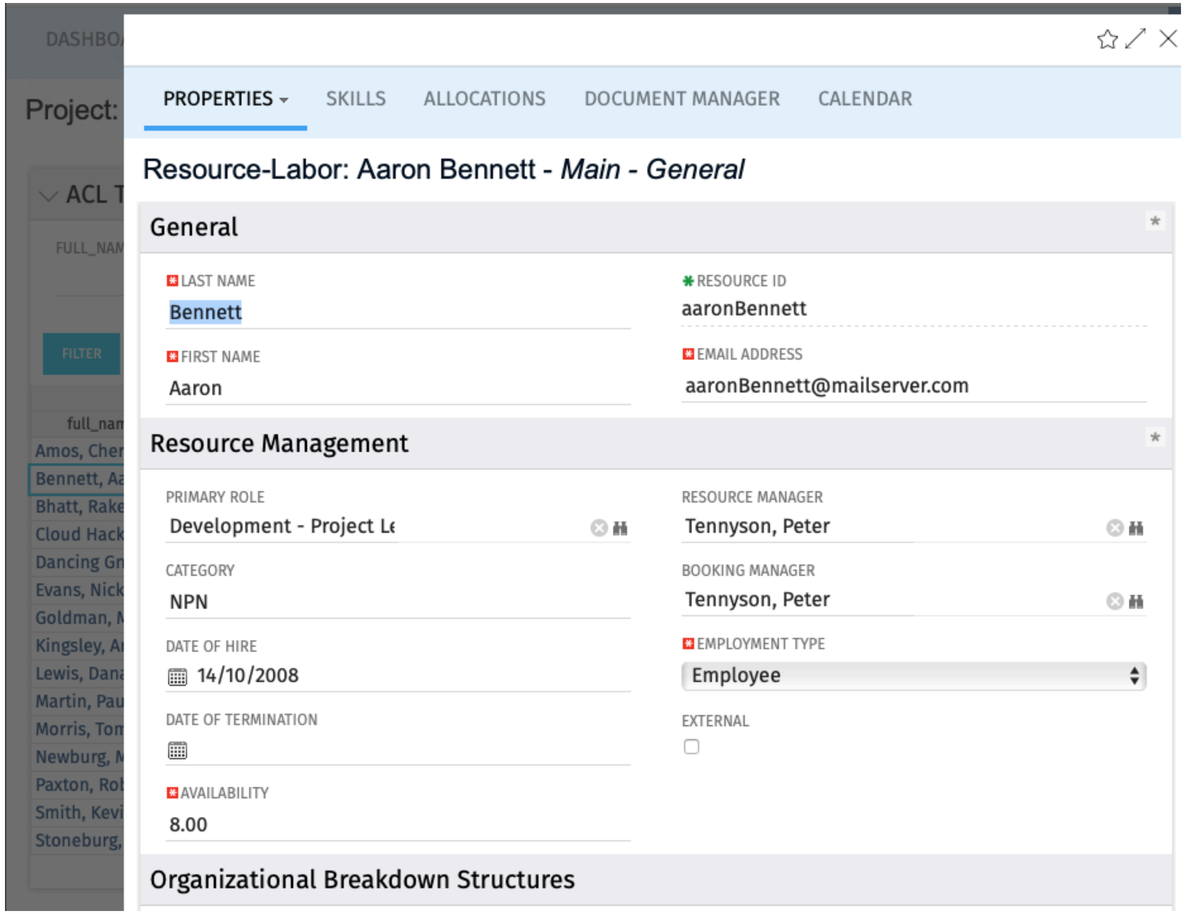
ACL Test 5

FULL_NAME

PARAM_ISROLE: No

	JUN 19	JUL 19	AUG 19
full_name	avail	avail	avail
Amos, Cheryl	160	184	176
Bennett, Aaron	160	184	176
Bhatt, Rakesh	160	184	176
Cloud Hackers	1,680	1,932	1,848
Dancing Gnomes	2,800	3,220	3,080
Evans, Nick	160	184	176
Goldman, Mark	160	184	176
Kingsley, Art	160	184	176
Lewis, Dana	160	184	176
Martin, Paul	160	184	176
Morris, Tom	160	184	176
Newburg, Mary	160	184	176
Paxton, Robyn	160	184	176
Smith, Kevin	160	184	176
Stoneburg, Sam	160	184	176

Displaying 1 - 15 of 15



Resource-Labor: Aaron Bennett - *Main* - *General*

General

LAST NAME: Bennett	RESOURCE ID: aaronBennett
FIRST NAME: Aaron	EMAIL ADDRESS: aaronBennett@mailserver.com

Resource Management

PRIMARY ROLE: Development - Project Le	RESOURCE MANAGER: Tennyson, Peter
CATEGORY: NPN	BOOKING MANAGER: Tennyson, Peter
DATE OF HIRE: 14/10/2008	EMPLOYMENT TYPE: Employee
DATE OF TERMINATION:	EXTERNAL: □
AVAILABILITY: 8.00	

Organizational Breakdown Structures

Security Annotation

@WHERE:SECURITY:PROJECT:**<entity_id>**@

@WHERE:SECURITY:RESOURCE:**<entity_id>**@

- Only Project and Resource annotation is supported
- Can only be used in WHERE clause
- Replaces annotation with SQL where clause

Security Annotation: Examples

```
SELECT
  @SELECT:R.ID:ID@,
  @SELECT:R.UNIQUE_NAME:UNAME@,
  @SELECT:R.FULL_NAME:FULLNAME@
FROM SRM_RESOURCES R
WHERE R.IS_ACTIVE = 1
  AND @WHERE:SECURITY:RESOURCE:R.ID@
  AND @FILTER@
```

```
SELECT
  @SELECT:DIM:USER_DEF:IMPLIED:RESOURCE:R.ID:ID@,
  @SELECT:DIM_PROP:USER_DEF:IMPLIED:RESOURCE:R.UNIQUE_NAME:UNAME@,
  @SELECT:DIM_PROP:USER_DEF:IMPLIED:RESOURCE:R.FULL_NAME:FULLNAME@
FROM SRM_RESOURCES R
WHERE R.IS_ACTIVE = 1
  AND @WHERE:SECURITY:RESOURCE:R.ID@
  AND @FILTER@
```

Other Annotations

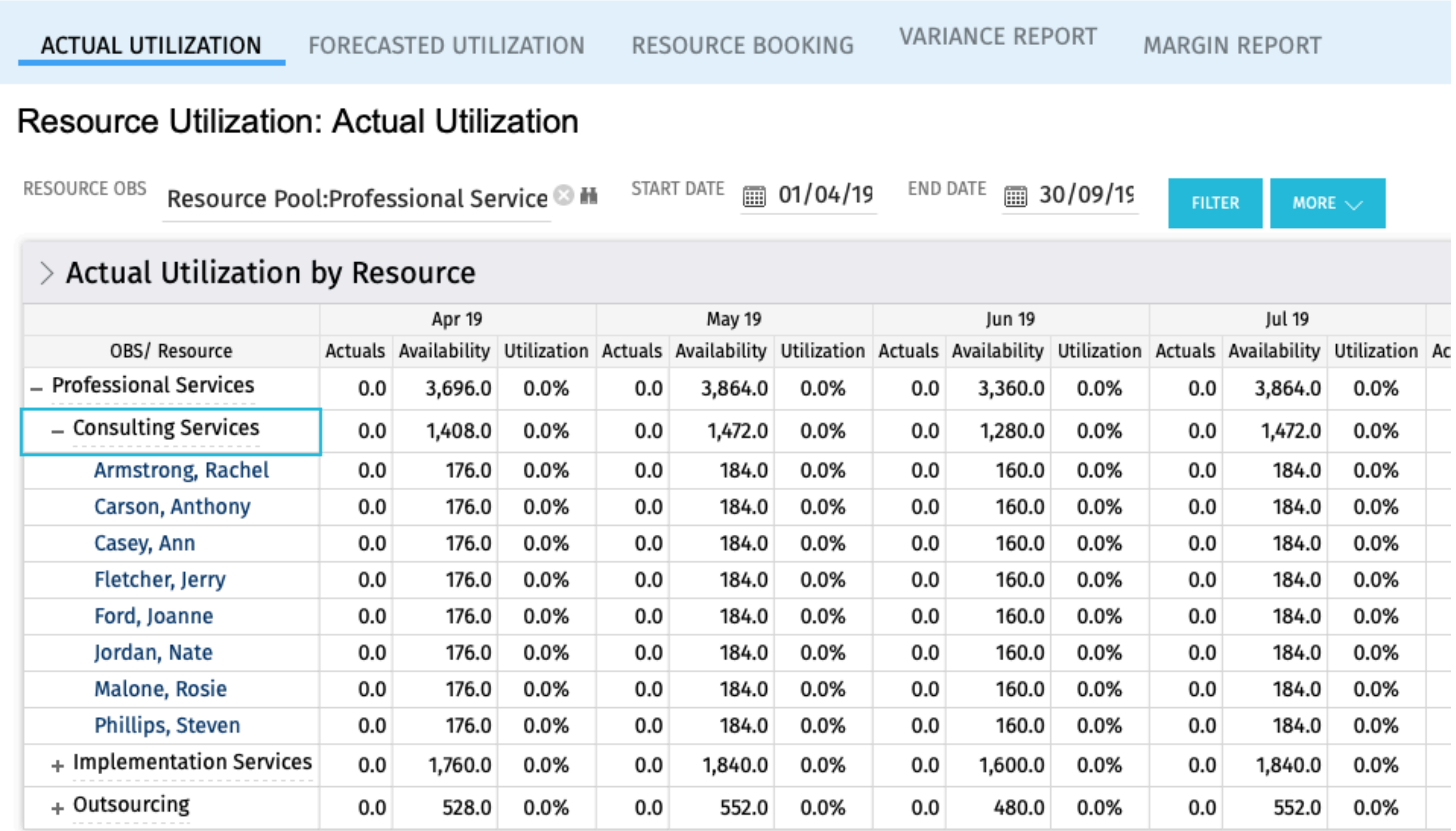
NSQL, still valuable even in the Modern UX

Dynamic Lookups: @BROWSE-ONLY

```
SELECT
  @SELECT:R.ID:ID@,
  @SELECT:R.UNIQUE_NAME:UNAME@,
  @SELECT:R.FULL_NAME:FULLNAME@
FROM SRM_RESOURCES R
WHERE
@BROWSE-ONLY:
  R.IS_ACTIVE = 1 AND
  @WHERE:SECURITY:RESOURCE:R.ID@ AND
:BROWSE-ONLY@
  @FILTER@
```

- In UI whilst browsing to select on active resources that you have rights to displayed.
- If subsequently, selected resource is inactivated, then UI would not display the value..
- With BROWSE-ONLY, SQL within the annotation is only included whilst browsing.

Hierarchical Queries



Resource Utilization: Actual Utilization

RESOURCE OBS Resource Pool:Professional Service

START DATE 01/04/19

END DATE 30/09/19

Actual Utilization by Resource

	Apr 19			May 19			Jun 19			Jul 19			
OBS/ Resource	Actuals	Availability	Utilization	Actuals	Availability	Utilization	Actuals	Availability	Utilization	Actuals	Availability	Utilization	Ac
– Professional Services	0.0	3,696.0	0.0%	0.0	3,864.0	0.0%	0.0	3,360.0	0.0%	0.0	3,864.0	0.0%	
– Consulting Services	0.0	1,408.0	0.0%	0.0	1,472.0	0.0%	0.0	1,280.0	0.0%	0.0	1,472.0	0.0%	
Armstrong, Rachel	0.0	176.0	0.0%	0.0	184.0	0.0%	0.0	160.0	0.0%	0.0	184.0	0.0%	
Carson, Anthony	0.0	176.0	0.0%	0.0	184.0	0.0%	0.0	160.0	0.0%	0.0	184.0	0.0%	
Casey, Ann	0.0	176.0	0.0%	0.0	184.0	0.0%	0.0	160.0	0.0%	0.0	184.0	0.0%	
Fletcher, Jerry	0.0	176.0	0.0%	0.0	184.0	0.0%	0.0	160.0	0.0%	0.0	184.0	0.0%	
Ford, Joanne	0.0	176.0	0.0%	0.0	184.0	0.0%	0.0	160.0	0.0%	0.0	184.0	0.0%	
Jordan, Nate	0.0	176.0	0.0%	0.0	184.0	0.0%	0.0	160.0	0.0%	0.0	184.0	0.0%	
Malone, Rosie	0.0	176.0	0.0%	0.0	184.0	0.0%	0.0	160.0	0.0%	0.0	184.0	0.0%	
Phillips, Steven	0.0	176.0	0.0%	0.0	184.0	0.0%	0.0	160.0	0.0%	0.0	184.0	0.0%	
+ Implementation Services	0.0	1,760.0	0.0%	0.0	1,840.0	0.0%	0.0	1,600.0	0.0%	0.0	1,840.0	0.0%	
+ Outsourcing	0.0	528.0	0.0%	0.0	552.0	0.0%	0.0	480.0	0.0%	0.0	552.0	0.0%	

Enables expansion of row and display of sub rows

Can have many levels

Hierarchical Queries

Select dimension property named **hg_has_children**

- Unique for all rows in the dimension
- If null – then does not show [+]
- Passed to the query on clicking the [+] as the **hg_row_id** parameter

@WHERE:PARAM:USER_DEF:STRING:hg_row_id@

- Set to value of clicked [+] row **hg_has_children** attribute
- Initial value on portlet display is NULL

Hierarchical Queries

@WHERE:PARAM:USER_DEF:STRING:hg_all_rows@

- Used for Export to Excel
- Set to 1 when exporting otherwise NULL
- Used to enable/disable sub levels from being exported

Hierarchical Queries

Common SQL Structure

```
SELECT @SELECT....
FROM
(
   SELECT <the parent Level>
   UNION
   SELECT <the first child level>
   UNION
   SELECT <the next child level>
   UNION
   ....
)
WHERE @FILTER@
```

hg_has_children must uniquely identify the data to retrieve or not retrieve in each of the sub-queries

Some examples can be found on the Resource Utilisation page – uses OBS

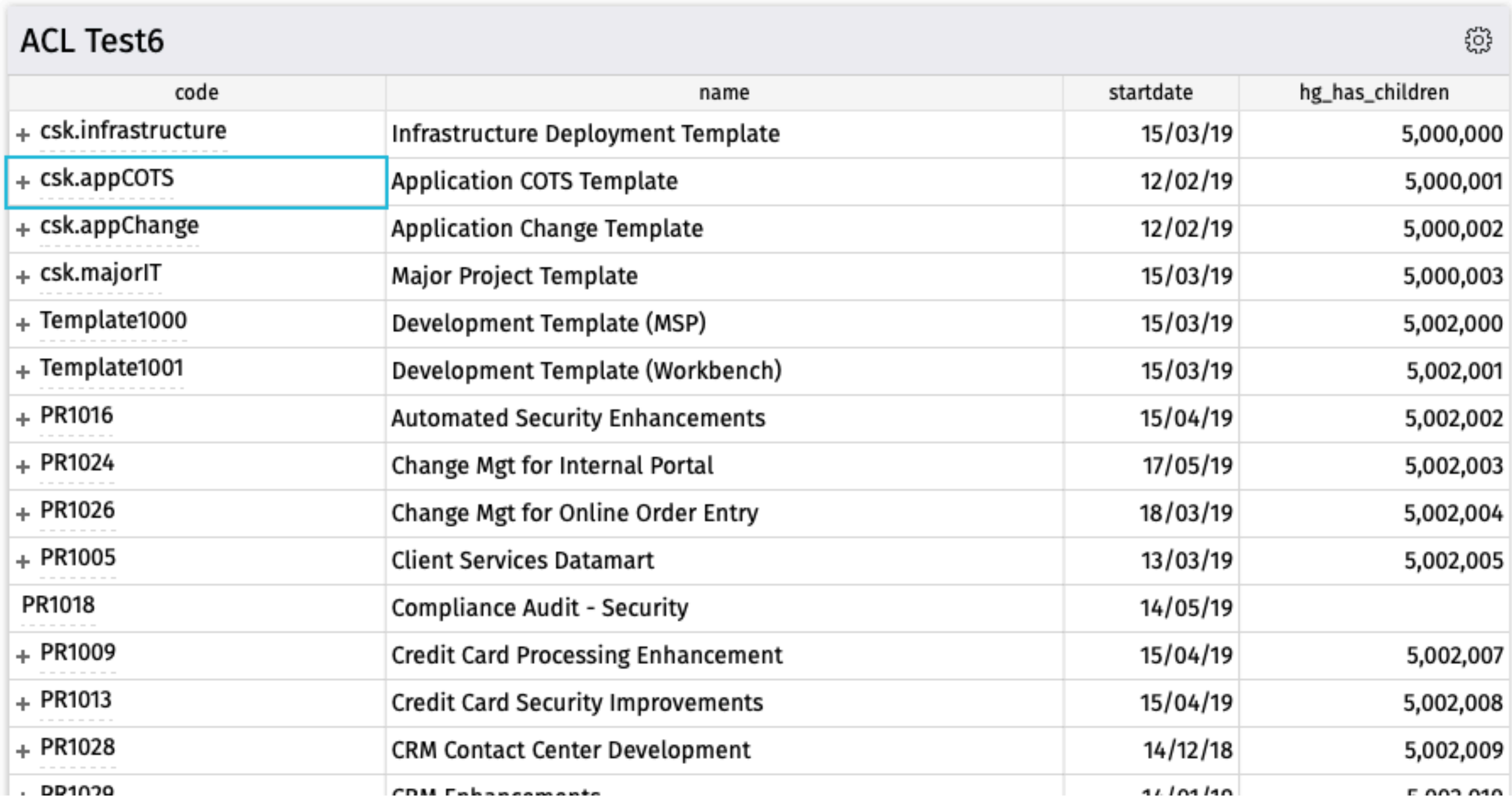
Hierarchical Queries

Example

```
SELECT @SELECT:DIM:USER_DEF:IMPLIED:DATA:ID:ID@,
  @SELECT:DIM_PROP:USER_DEF:IMPLIED:DATA:CODE:CODE@,
  @SELECT:DIM_PROP:USER_DEF:IMPLIED:DATA:NAME:NAME@,
  @SELECT:DIM_PROP:USER_DEF:IMPLIED:DATA:STARTDATE:STARTDATE@,
  @SELECT:DIM_PROP:USER_DEF:IMPLIED:DATA:HG_HAS_CHILDREN:HG_HAS_CHILDREN@
FROM (
SELECT I.ID ID, I.CODE CODE, I.NAME NAME, I.SCHEDULE_START STARTDATE,
  MAX(CASE WHEN T.PRID IS NULL THEN NULL ELSE I.ID END) HG_HAS_CHILDREN
FROM INV_INVESTMENTS I
  LEFT OUTER JOIN PRTASK T ON I.ID = T.PRPROJECTID
WHERE I.ODF_OBJECT_CODE = 'project'
  AND @WHERE:PARAM:USER_DEF:STRING:HG_ROW_ID@ IS NULL
GROUP BY I.ID, I.CODE, I.NAME, I.SCHEDULE_START
UNION
SELECT T.PRID ID, T.PREXTERNALID CODE, T.PRNAME NAME, T.PRSTART STARTDATE,
  NULL HG_HAS_CHILDREN
FROM PRTASK T
WHERE @WHERE:PARAM:USER_DEF:STRING:HG_ROW_ID@ IS NOT NULL
  AND T.PRPROJECTID=@WHERE:PARAM:USER_DEF:STRING:HG_ROW_ID@
)
WHERE @FILTER@
```

Hierarchical Queries

Example

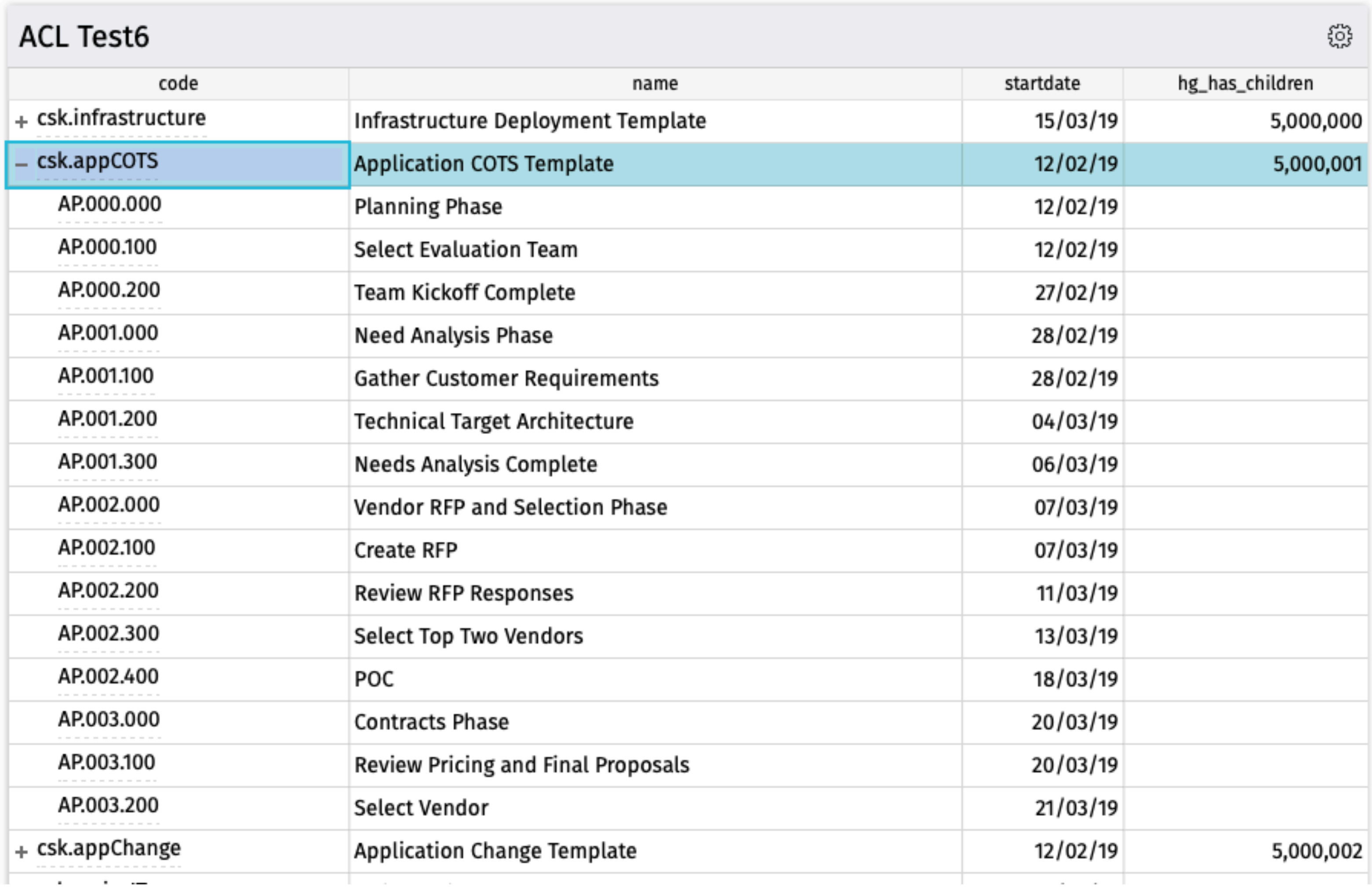


ACL Test6

code	name	startdate	hg_has_children
+ csk.infrastructure	Infrastructure Deployment Template	15/03/19	5,000,000
+ csk.appCOTS	Application COTS Template	12/02/19	5,000,001
+ csk.appChange	Application Change Template	12/02/19	5,000,002
+ csk.majorIT	Major Project Template	15/03/19	5,000,003
+ Template1000	Development Template (MSP)	15/03/19	5,002,000
+ Template1001	Development Template (Workbench)	15/03/19	5,002,001
+ PR1016	Automated Security Enhancements	15/04/19	5,002,002
+ PR1024	Change Mgt for Internal Portal	17/05/19	5,002,003
+ PR1026	Change Mgt for Online Order Entry	18/03/19	5,002,004
+ PR1005	Client Services Datamart	13/03/19	5,002,005
PR1018	Compliance Audit - Security	14/05/19	
+ PR1009	Credit Card Processing Enhancement	15/04/19	5,002,007
+ PR1013	Credit Card Security Improvements	15/04/19	5,002,008
+ PR1028	CRM Contact Center Development	14/12/18	5,002,009
+ PR1029	CRM Enhancements	14/01/19	5,002,010

Hierarchical Queries

Example



ACL Test6

code	name	startdate	hg_has_children
+ csk.infrastructure	Infrastructure Deployment Template	15/03/19	5,000,000
– csk.appCOTS	Application COTS Template	12/02/19	5,000,001
AP.000.000	Planning Phase	12/02/19	
AP.000.100	Select Evaluation Team	12/02/19	
AP.000.200	Team Kickoff Complete	27/02/19	
AP.001.000	Need Analysis Phase	28/02/19	
AP.001.100	Gather Customer Requirements	28/02/19	
AP.001.200	Technical Target Architecture	04/03/19	
AP.001.300	Needs Analysis Complete	06/03/19	
AP.002.000	Vendor RFP and Selection Phase	07/03/19	
AP.002.100	Create RFP	07/03/19	
AP.002.200	Review RFP Responses	11/03/19	
AP.002.300	Select Top Two Vendors	13/03/19	
AP.002.400	POC	18/03/19	
AP.003.000	Contracts Phase	20/03/19	
AP.003.100	Review Pricing and Final Proposals	20/03/19	
AP.003.200	Select Vendor	21/03/19	
+ csk.appChange	Application Change Template	12/02/19	5,000,002

Hierarchical Queries

Advice

Filtering: Implement filters using Parameters

Export: When using hg_all_rows, export does not sort rows based on the hierarchy

Layout: All levels must have the same field layout

Sorting: All levels use the same field sorting so add a column for sort order

NSQL, still valuable even in the Modern UX

Questions?

Thank you for attending

NSQL, still valuable even in the Modern UX

Phone

+44 844 736 2500

Email

ppmacademy@pemari.com

Website

www.pemari.com

Let us know how we can improve!
Don't forget to fill out the feedback forms!