SIGCONNECT
Higher Education Conference

# Degree Works Reporting

Dennis Crowson, Ph.D. Senior Consultant

# Housekeeping

All participants are automatically muted by webinar administrators

Please type any questions using the chat feature on the right-hand side of your screen

Webinar will be recorded for future reference on the SIGConnect website

@SIGCorpLive

# PRESENTER

## Dennis K. Crowson

Senior Consultant

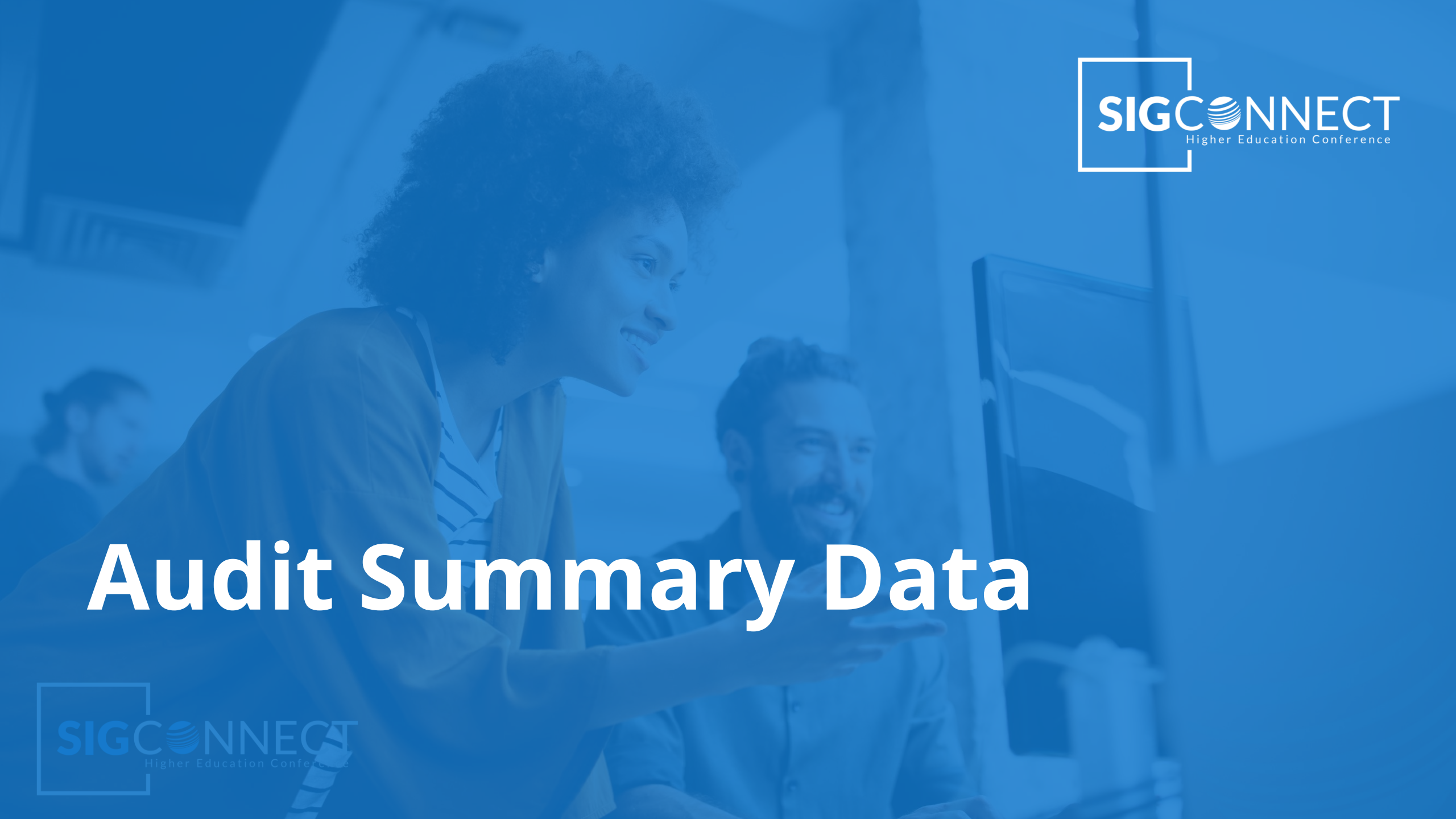
- 35+ years working in higher education
- 25 years experience with Degree Works
- Degree Works Technical Consultant for SunGard Higher Education (now Ellucian)
- Ph.D. in Higher Education Administration
- Former Vice President Student Services, Dean of Enrollment, and Registrar at Blinn College

# Degree Works Reporting

- Degree Works contains a great deal of data about student audits and plans
- How do I get to this data?
- What is out there?
  - How do I know if a student is ready to graduate?
  - Are there students that we are missing who are eligible to graduate?
  - What classes are still needing?
  - What classes are students planning on taking?

# Let's Start with Basics

- DW stores data mainly in two sets of tables:
  - Tables that contain summary information on about the Audit
  - In depth information about what is going on with the audit
    - CPA (Curriculum Planning Assistant) data



# Audit Summary Data

# Audit Summary Data

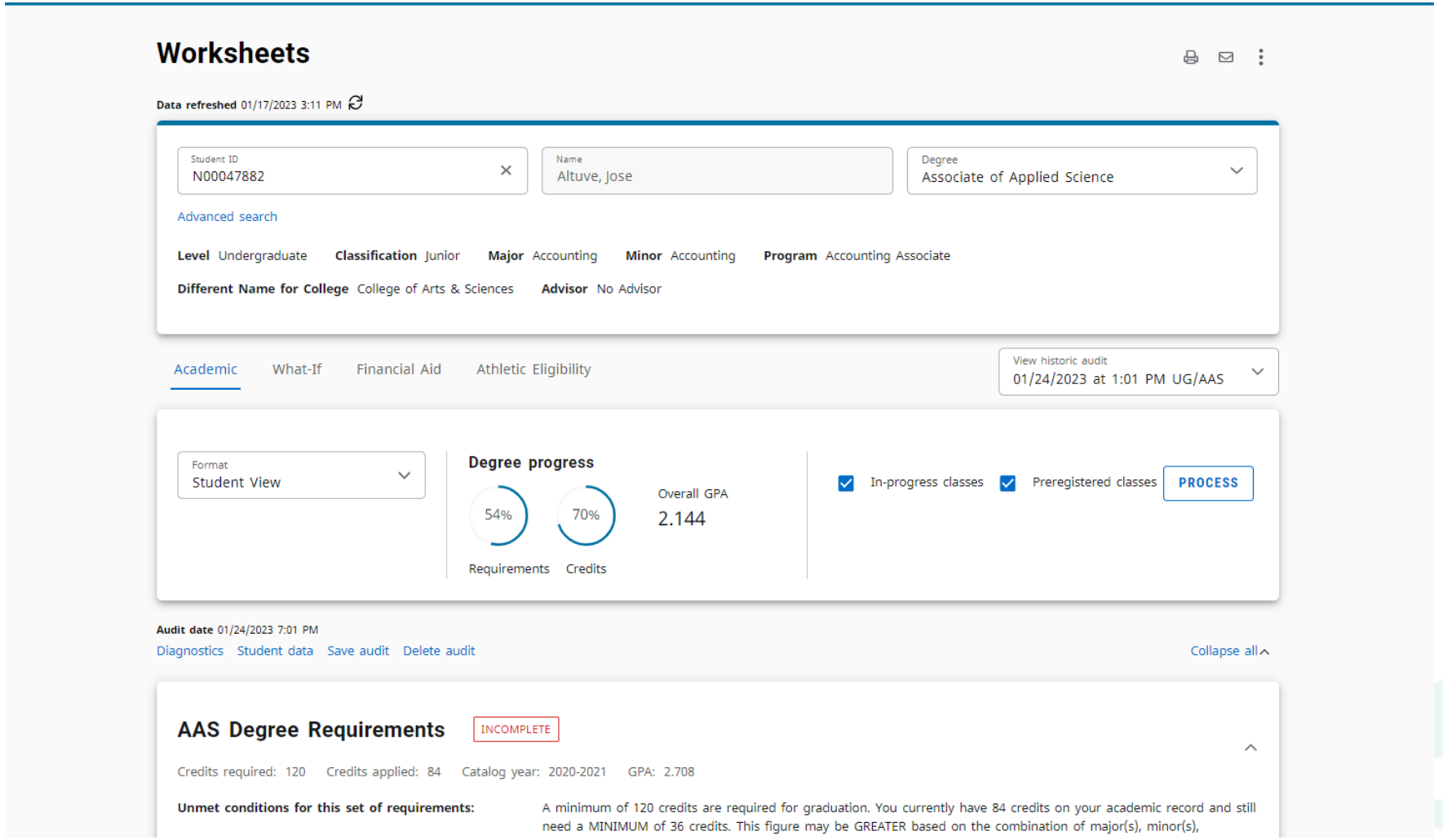
- Every time a new audit is run on a student a record is created and stored in the **DAP_AUDIT_DTL** table

- CFG020/DAP14 contains a setting to determine the maximum number of **DAP_AUDIT_DTL** records to keep per student per degree
  - Recommend this not be set to > 3
  - Frozen audits are not counted in this number

# DAP_AUDIT_DTL

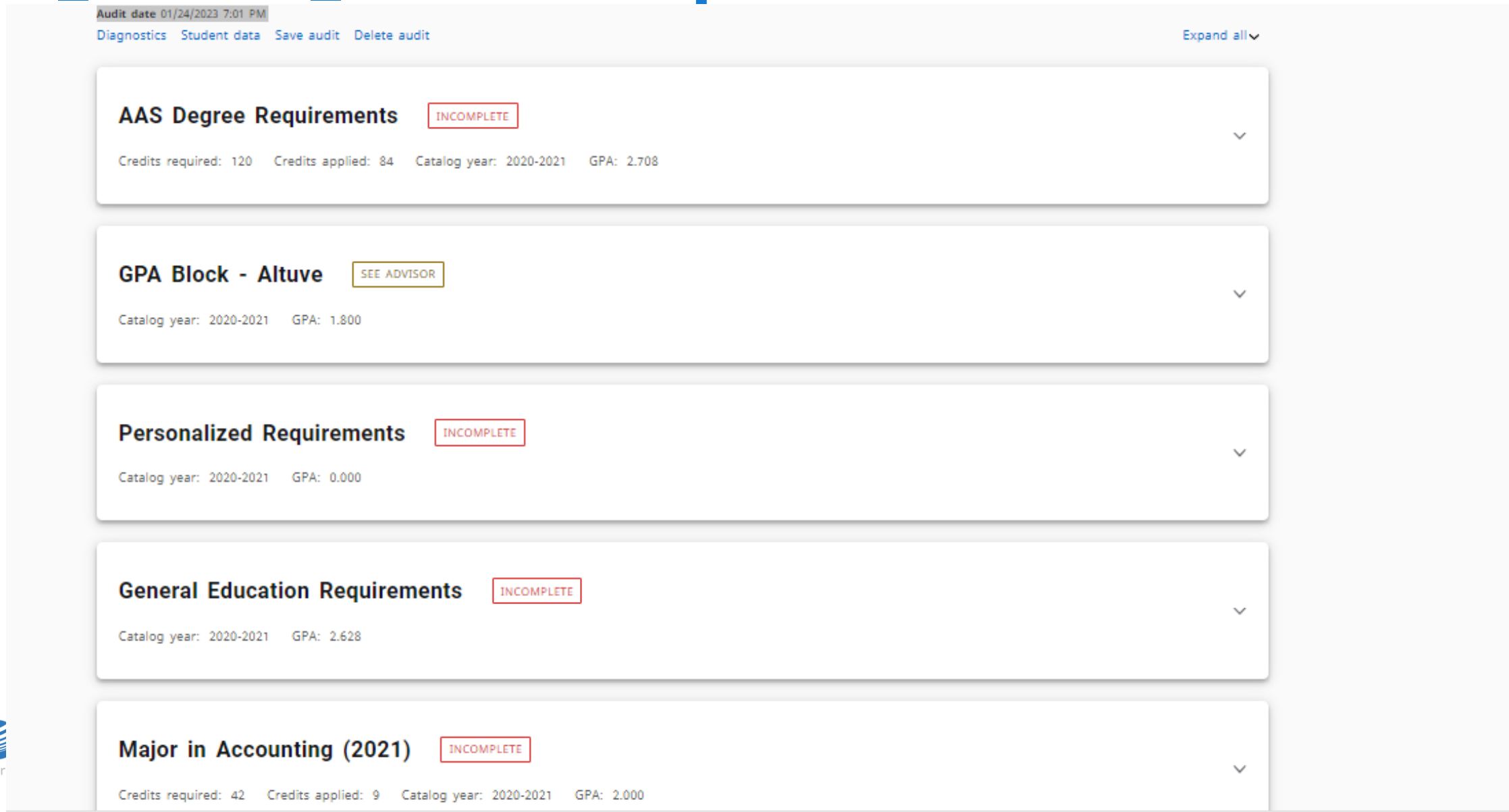
- Contains good summary information on the student's audit
- For each block in the audit a record containing the following is stored:
  - Block type (DEGREE, MAJOR, MINOR, CONC, OTHER)
  - Block value (AAS, BUSI, etc. )
  - Percent complete for the block
  - GPA for the block
- There are up to ten sets of these block groupings

# Example




SIGCONNECT Higher Education Conference

# DAP_AUDIT_DTL Example



# DAP_AUDIT_DTL

| Field | Description | Value |
|---|---|---|
| DAP_AUD_BLOCK1 | Usually is the student's Degree block | DEGREE |
| DAP_AUD_VALUE1 | Value of student's top block | AAS |
| DAP_AUD_PCT1 | Percent complete of the top block | 53 |
| DAP_AUD_GPA1 | GPA of the top block | 2.708 |
| DAP_AUD_BLOCK2 | The type of block for the student's SECOND block | MAJOR |
| DAP_AUD_VALUE2 | Value of the SECOND block | ACCT |
| DAP_AUD_PCT2 | Percent complete of the second block | 37 |
| DAP_AUD_GPA2 | GPA of the second block | 2 |

# Reporting % Complete

- How many students are > 50% complete using **DAP_AUDIT_DTL**

```
Select *
from DAP_AUDIT_DTL
Where DAP_AUD_PCT1 > 50;
```

# Reporting % Complete

```
select *
from dap_audit_dtl
where dap_aud_pct1>50;
```

y Result

SQL | All Rows Fetched: 12 in 0.064 seconds

| DAP_STU_ID | DAP_SCHOOL | DAP_DEGREE | DAP_STU_LEVEL | DAP_AUDIT_DATE | DAP_AUDIT_TIME | DAP_AUDIT_TYPE | DAP_FR... | DAP_AU... | DAP_AUDIT_PCT | DAP_AUDIT_GPA | DAP_AUD_BLOCK1 | DAP_A... | DAP_AUD_PCT1 | DAP_AUD_GPA1 | DAP_ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| N00010099 | UG | BA | SR | ..08-FEB-22 | 215812 | AA | | ... | 99 | 0 | DEGREE | BA ... | 99 | 0 | MAJOR |
| N00018934 | UG | AA | JR | ..03-FEB-22 | 125337 | AA | | ... | 55 | 3.464 | DEGREE | AA ... | 55 | 3.464 | MAJOR |
| N00018934 | UG | AA | JR | ..08-NOV-21 | 140449 | AA | | ... | 55 | 3.638 | DEGREE | AA ... | 55 | 3.638 | MAJOR |
| N00018934 | UG | AA | JR | ..08-NOV-21 | 104032 | AA | | ... | 55 | 3.638 | DEGREE | AA ... | 55 | 3.638 | MAJOR |
| N00047884 | UG | BS | FR | ..16-AUG-22 | 082600 | AA | | 22/23 ... | 56 | 1.5 | DEGREE | BS ... | 56 | 1.5 | MAJOR |
| N00047882 | UG | AAS | JR | ..20-JAN-23 | 154529 | AA | | ... | 53 | 2.708 | DEGREE | AAS ... | 53 | 2.708 | MAJOR |
| N00018934 | UG | AA | JR | ..09-JUN-22 | 090635 | AE | | ... | 55 | 3.66 | DEGREE | AA ... | 55 | 3.66 | MAJOR |
| N00018934 | UG | AA | JR | ..10-JAN-23 | 081804 | AA | | ... | 60 | 3.535 | DEGREE | AA ... | 60 | 3.535 | MAJOR |
| N00018934 | UG | AA | JR | ..02-SEP-22 | 094240 | AA | | ... | 60 | 3.535 | DEGREE | AA ... | 60 | 3.535 | MAJOR |
| N00047884 | UG | BS | FR | ..16-AUG-22 | 082602 | AA | | 22/23 ... | 56 | 1.5 | DEGREE | BS ... | 56 | 1.5 | MAJOR |
| N00047882 | UG | AAS | JR | ..24-JAN-23 | 190139 | AA | | ... | 53 | 2.708 | DEGREE | AAS ... | 53 | 2.708 | MAJOR |
| N00018934 | UG | AA | JR | ..02-SEP-22 | 094215 | AA | | ... | 55 | 3.685 | DEGREE | AA ... | 55 | 3.685 | MAJOR |

*Remember since up to 3 audits, probably want to add a where clause to grab the latest audit

# DAP_AUDIT_DTL

**Things to be aware of and to look out for when reporting from DAP_AUDIT_DTL:**

- Multiple records per student
  - ❑ Should delimit by the latest audit
  - ❑ Separate records per **degree**, school (level), audit type, and active term
- Frozen audits – do you want to report?
- Watch out for results in the 2 – 10 fields
  - ❑ Cannot always guarantee what these represent
- Limitations
  - ❑ Does not include Credits Needed or Completed
  - ❑ Does not contain RA#

SIGCONNECT
Higher Education Conference

# CPA Data

# Curriculum Planning Assistant (CPA)

**DAP_AUDIT_DTL** is limited on what it contains

- Often you need more in-depth data such as number of credits needed or used in a block, classes needed, classes used, etc.
- Must use CPA to get this more in-depth information
- CPA is a terrible acronym!! Really “Additional or In-depth DW Data”

# CPA Data

- By default, DW audits are stored in a **binary tree format** (dap_audtree_dtl) so most of the DW audit data cannot be queried directly.
- An external process must take these audit trees and extract them into database tables for reporting.
- UCX CFG020 / RESULTS controls what type of data will get built.

# CPA Data

- The DW CPA data is built by specifying the items you want to build and then running a process to create the CPA results records.
- CPA data can include almost any data that you see on an audit
- CPA tables include:
  - **DAP_RESULT_DTL**
    - Data within this table is based on field *DAP_RESULT_TYPE*
  - **DAP_RESCLASS_DTL**
  - DAP_NONRES_DTL

# Steps to Create CPA Data

Mark values in CFG020/Results to report on

- Advanced Reporting Manual is a good reference

Run CPA Process - DAP25

- If resjobs is turned on will run in background
- WebTransit

Records are created in the table *DAP_RESULT_DTL.* These records based on *DAP_RESULT_TYPE* field in the table

# DAP_RESULT_DTL

**DAP_RESULT_DTL is the main table for CPA data**

- Only one set of results per:
  - Student ID
  - Audit Type (usually AA)
  - Level
  - Degree
  - Active Term

- Each set of results gets re-written each time the CPA process is run

# DAP_RESULT_DTL

## Structure of DAP_RESULT_DTL

- DAP_STU_ID
- DAP_AUDIT_TYPE – Usually AA for Academic Audit
- DAP_SCHOOL
- DAP_DEGREE
- DAP_ACTIVE_TERM
- **DAP_RESULT_TYPE** – Value and Number fields are dependent on this type
- DAP_VALUE1…VALUE4
- DAP_NUMBER1…NUMBER4
- DAP_FREETEXT

# DAP_RESCLASS_DTL

**DAP_RESCLASS_DTL**

- Records are created in this table of the classes a student has taken
- Similar to a student's class history

**Structure of DAP_RESCLASS_DTL**

- DAP_STU_ID, DAP_AUDIT_TYPE, DAP_SCHOOL, DAP_DEGREE, DAP_ACTIVE_TERM
- DAP_CLASS_ID
- DAP_DISCIPLINE
- DAP_COURSE_NUM
- DAP_COURSE_TITLE
- DAP_TERM
- DAP_GRADE
- Etc.

# DAP_RESULT_DTL – DAP_RESULT_TYPE

- Within **DAP_RESULT_DTL**, the **DAP_RESULT_TYPE** field determines what the rest of the values in the record will represent
- Examples of DAP_RESULT_TYPE records include:
  - ❑ AUDITID
  - ❑ BLOCKGPA
  - ❑ CLASSAPPLIED
  - ❑ BLOCKCRCLNEEDED
  - ❑ BLOCKCRCLAPPLIED

# Example – Credits Needed

## Example – how many credits are still needed to complete a student's degree?

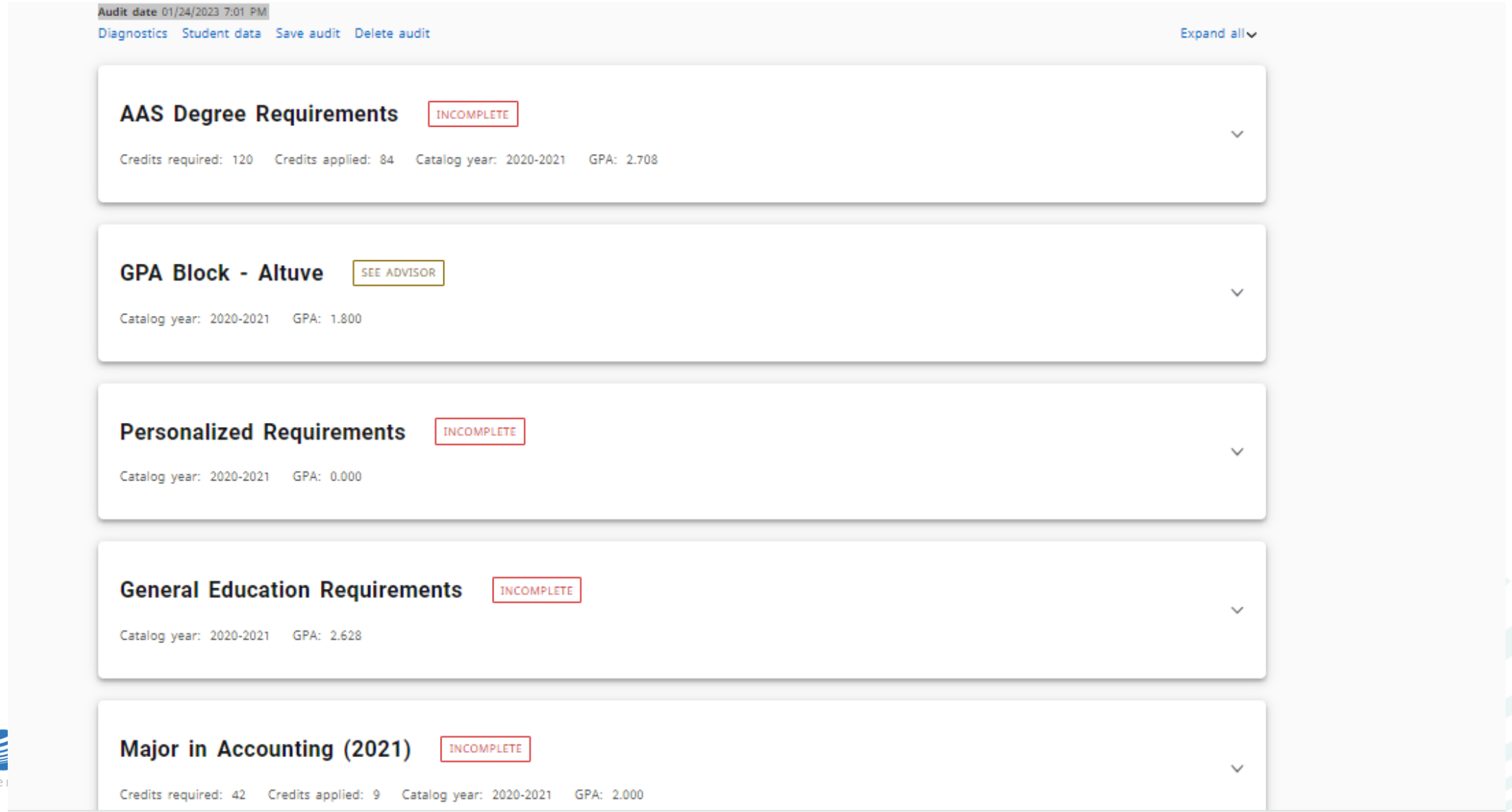
- Once the CPA data has been created with the appropriate flags set in CFG020/Results can use the records created to find how many credits are needed to complete a degree

- Will be querying **DAP_RESULT_DTL** looking for **DAP_RESULT_TYPE = 'BLOCKCRCLNEEDED'**

# BLOCKCRCLNEEDED

| Field Name | Description | Example |
|---|---|---|
| DAP_REQ_ID | Requirement ID of the block | RA000123 |
| DAP_RESULT_TYPE | Determination of what rest of the fields represent | BLOCKCRCLNEEDED |
| DAP_VALUE1 | Credits Needed (must have a header credit requirement) | 43 |
| DAP_VALUE2 | Classes Needed (must have a header class requirement | 15 |
| Rest of fields | NOT USED | |

# DAP_REPORT_DTL Example



# Example – How many credits does Jose Altuve need to graduate with AAS?

```
Select DAP_REQ_ID as “Block”
, DAP_VALUE1 as “Credits Needed”
From DAP_RESULT_DTL
Where DAP_STU_ID = ‘N00047882’
And DAP_RESULT_TYPE = ‘BLOCKCRCLNEEDED’
And DAP_DEGREE = ‘AAS’
And DAP_REQ_ID = 'RA000456';
```

# Example – Credits Needed

```
1  select
2  a.dap_req_id as "Requirement"
3  , a.dap_valuel as "Credits Needed"
4  from dap_result_dtl a
5  where
6    a.dap_stu_id = 'N00047882'
7    and a.dap_result_type = 'BLOCKCRCLNEEDED'
8    and a.dap_degree = 'AAS'
9    and a.dap_req_id = 'RA000456'
10   ;
```

Script Output | Query Result

SQL | All Rows Fetched: 1 in 0.072 seconds

| | Requirement | Credits Needed |
|---|---|---|
| 1 | RA000456 | 39 |

# Expanded Example – What Blocks are these?

- **Now let's improve the report to show all the blocks' information including the block titles and % completed per block**

- Blocks are defined in the table **DAP_REQ_BLOCK**

- % Complete could come from **DAP_AUDIT_DTL**, but for this example, let's get that from **DAP_RESULT_DTL** also

# Requirement Blocks

- Scribe block information is stored in **DAP_REQ_BLOCK**
- Scribe code is stored in *CLOB*

| | COLUMN_NAME | DATA_TYPE | NULLABLE | DATA_DEFAULT | COLUMN_ID | COMMENTS |
|---|---|---|---|---|---|---|
| 1 | REQUIREMENT_ID | VARCHAR2(8 BYTE) | No | (null) | 1 | (null) |
| 2 | BLOCK_TYPE | VARCHAR2(12 BYTE) | No | (null) | 2 | (null) |
| 3 | BLOCK_VALUE | VARCHAR2(12 BYTE) | No | (null) | 3 | (null) |
| 4 | TITLE | VARCHAR2(50 BYTE) | Yes | (null) | 4 | (null) |
| 5 | PERIOD_START | VARCHAR2(12 BYTE) | Yes | (null) | 5 | (null) |
| 6 | PERIOD_STOP | VARCHAR2(12 BYTE) | Yes | (null) | 6 | (null) |
| 7 | SCHOOL | VARCHAR2(12 BYTE) | Yes | (null) | 7 | (null) |
| 8 | DEGREE | VARCHAR2(12 BYTE) | Yes | (null) | 8 | (null) |
| 9 | COLLEGE | VARCHAR2(12 BYTE) | Yes | (null) | 9 | (null) |
| 10 | MAJOR1 | VARCHAR2(12 BYTE) | Yes | (null) | 10 | (null) |
| 11 | MAJOR2 | VARCHAR2(12 BYTE) | Yes | (null) | 11 | (null) |
| 12 | CONCENTRATION | VARCHAR2(12 BYTE) | Yes | (null) | 12 | (null) |
| 13 | MINOR | VARCHAR2(12 BYTE) | Yes | (null) | 13 | (null) |
| 14 | LIBERAL_LEARNING | VARCHAR2(12 BYTE) | Yes | (null) | 14 | (null) |
| 15 | SPECIALIZATION | VARCHAR2(12 BYTE) | Yes | (null) | 15 | (null) |
| 16 | PROGRAM | VARCHAR2(12 BYTE) | Yes | (null) | 16 | (null) |
| 17 | STUDENT_ID | VARCHAR2(10 BYTE) | Yes | (null) | 17 | (null) |
| 18 | PARSE_STATUS | VARCHAR2(2 BYTE) | Yes | (null) | 18 | (null) |
| 19 | PARSE_DATE | DATE | Yes | (null) | 19 | (null) |
| 20 | PARSE_WHO | VARCHAR2(14 BYTE) | Yes | (null) | 20 | (null) |
| 21 | PARSE_WHAT | VARCHAR2(30 BYTE) | Yes | (null) | 21 | (null) |
| 22 | LOCK_VERSION | FLOAT | Yes | 1 | 22 | (null) |
| 23 | REQUIREMENT_TEXT | CLOB | No | (null) | 23 | (null) |
| 24 | CREATE_DATE | DATE | Yes | (null) | 24 | (null) |
| 25 | CREATE_WHO | VARCHAR2(14 BYTE) | Yes | (null) | 25 | (null) |
| 26 | CREATE_WHAT | VARCHAR2(30 BYTE) | Yes | (null) | 26 | (null) |
| 27 | MODIFY_DATE | DATE | Yes | (null) | 27 | (null) |
| 28 | MODIFY_WHO | VARCHAR2(14 BYTE) | Yes | (null) | 28 | (null) |
| 29 | MODIFY_WHAT | VARCHAR2(30 BYTE) | Yes | (null) | 29 | (null) |

# Credits needed with Block Titles

```
select dap_req_id as "Block“
, title as "Block Title"
, dap_value1 as "Credits Needed"
from dap_result_dtl
join dap_req_block on
  dap_req_id = requirement_id
where dap_stu_id = 'N00047882'
and dap_result_type='BLOCKCRCLNEEDED'
and dap_degree = 'AAS'
;
```

```
1  select
2  a.dap_req_id as "Requirement"
3  , title as "Block"
4  , a.dap_value1 as "Credits Needed"
5  from dap_result_dtl a
6  join dap_req_block on a.dap_req_id = requirement_id
7  where
8    a.dap_stu_id = 'N00047882'
9    and a.dap_result_type = 'BLOCKCRCLNEEDED'
0    and a.dap_degree = 'AAS'
1    ;
```

Query Result

SQL | All Rows Fetched: 4 in 0.211 seconds

| | Requirement | Block | Credits Needed |
|---|---|---|---|
| 1 | RA000456 | AAS Degree Requirements | 39 |
| 2 | RA000395 | Accounting | 3 |
| 3 | RA000736 | Major in Accounting (2021) | 33 |
| 4 | RA000278 | General Education Requirements | 6 |

# Now add % Complete

- Requires an additional select into **DAP_RESULT_DTL**
- Will look for **DAP_RESULT_TYPE = 'BLOCKGPA'**
- Will tie the original selection of DAP_RESULT_DTL (a) with the new selection (b) on
  - Student ID
  - Degree
  - Active Term
  - Requirement ID

# Adding % Complete

Worksheet | Query Builder

```
1  select
2  a.dap_req_id as "Requirement"
3  , title as "Block"
4  , a.dap_valuel as "Credits Needed"
5  , b.dap_valuel as "Percent Complete"
6  from dap_result_dtl a
7  join dap_req_block on a.dap_req_id = requirement_id
8  join dap_result_dtl b on
9     a.dap_stu_id = b.dap_stu_id
10    and a.dap_degree=b.dap_degree
11    and a.dap_req_id = b.dap_req_id
12    and a.dap_active_term = b.dap_active_term
13    and b.dap_result_type = 'BLOCKGPA'
14 where
15   a.dap_stu_id = 'N00047882'
16   and a.dap_result_type = 'BLOCKCRCLNEEDED'
17   and a.dap_degree = 'AAS'
18   ;
```

Query Result

SQL | All Rows Fetched: 4 in 0.103 seconds

| | Requirement | Block | Credits Needed | Percent Complete |
|---|---|---|---|---|
| 1 | RA000456 | AAS Degree Requirements | 39 | 62.947369 |
| 2 | RA000395 | Accounting | 3 | 75 |
| 3 | RA000736 | Major in Accounting (2021) | 33 | 37.25 |
| 4 | RA000278 | General Education Requirements | 6 | 79.599998 |

# Adding Classes Needed

- Insert another join to DAP_RESULT_DTL
- Now looking for **DAP_RESULT_TYPE = 'CLASSNEEDED'**
- Outer join since some blocks may not need anything

```
select
a.dap_req_id as "Requirement"
, title as "Block"
, a.dap_valuel as "Credits Needed"
, b.dap_valuel as "Percent Complete"
, rtrim(c.dap_freetext) as "Class Needed"
from dap_result_dtl a
join dap_req_block on a.dap_req_id = requirement_id
join dap_result_dtl b on
   a.dap_stu_id = b.dap_stu_id
   and a.dap_degree=b.dap_degree
   and a.dap_req_id = b.dap_req_id
   and a.dap_active_term = b.dap_active_term
   and b.dap_result_type = 'BLOCKGPA'
left outer join dap_result_dtl c  on
   a.dap_stu_id = c.dap_stu_id
   and a.dap_degree=c.dap_degree
   and a.dap_req_id = c.dap_req_id
   and a.dap_active_term = c.dap_active_term
   and c.dap_result_type = 'CLASSNEEDED'
where
  a.dap_stu_id = 'N00047882'
  and a.dap_result_type = 'BLOCKCRCLNEEDED'
  and a.dap_degree = 'AAS'
  order by a.dap_req_id
  ;
```

uery Result

SQL | All Rows Fetched: 23 in 0.133 seconds

| | Requirement | Block | Credits Needed | Percent Complete | Class Needed |
|---|---|---|---|---|---|
| 1 | RA000278 | General Education Requirements | 6 | 79.599998 | INDI 110 |
| 2 | RA000278 | General Education Requirements | 6 | 79.599998 | HIST 2010 |
| 3 | RA000278 | General Education Requirements | 6 | 79.599998 | HIST 1493 |
| 4 | RA000278 | General Education Requirements | 6 | 79.599998 | HIST 225 |
| 5 | RA000278 | General Education Requirements | 6 | 79.599998 | HIST 201H |
| 6 | RA000278 | General Education Requirements | 6 | 79.599998 | HIST 201 |
| 7 | RA000278 | General Education Requirements | 6 | 79.599998 | POLS 2250 |
| 8 | RA000278 | General Education Requirements | 6 | 79.599998 | POLS 1113 |
| 9 | RA000278 | General Education Requirements | 6 | 79.599998 | POLS 1100 |
| 10 | RA000278 | General Education Requirements | 6 | 79.599998 | POLS 102 |
| 11 | RA000278 | General Education Requirements | 6 | 79.599998 | POLS 101 |
| 12 | RA000278 | General Education Requirements | 6 | 79.599998 | HIST 2020 |
| 13 | RA000395 | Accounting | 3 | 75 | ACCT 3@ |
| 14 | RA000395 | Accounting | 3 | 75 | ACCT 102 |
| 15 | RA000395 | Accounting | 3 | 75 | ACCT 1@ |

# Prettying up a bit

- Fill in name from **RAD_PRIMARY_MST**
- Make list of still needed courses aggregated

```
select  RTRIM(rad_name) as "Name", a.dap_req_id as "Requirement", title as "Block", a.dap_value1
as "Credits Needed", b.dap_value1 as "Percent Complete", NVL(LISTAGG(RTRIM(c.dap_value1)||'
'||RTRIM(c.dap_value2),', '),' ') as "Still Needed"
from dap_result_dtl a join rad_primary_mst on a.dap_stu_id = rad_id
join dap_req_block on a.dap_req_id = requirement_id
join dap_result_dtl b on
   a.dap_stu_id = b.dap_stu_id  and a.dap_degree=b.dap_degree  and a.dap_req_id =
b.dap_req_id  and a.dap_active_term = b.dap_active_term  and b.dap_result_type = 'BLOCKGPA'
left outer join dap_result_dtl c  on  a.dap_stu_id = c.dap_stu_id  and
a.dap_degree=c.dap_degree  and a.dap_req_id = c.dap_req_id  and a.dap_active_term =
c.dap_active_term  and c.dap_result_type = 'CLASSNEEDED'
where  a.dap_stu_id = 'N00047882' and a.dap_result_type = 'BLOCKCRCLNEEDED'  and
a.dap_degree = 'AAS'
 group by rad_name, a.dap_req_id, title, a.dap_value1, b.dap_value1  order by rad_name,
a.dap_req_id ;
```

# Prettying up a bit

| | Name | Requirement | Block | Credits Needed | Percent Complete | Still Needed |
|---|---|---|---|---|---|---|
| 1 | Altuve, Jose | RA000278 | General Education Requirements | 6 | 79.599998 | INDI 110, POLS 101, POLS 102, POLS 1100, POLS 1113, POLS 2250, HIST 201, H |
| 2 | Altuve, Jose | RA000395 | Accounting | 3 | 75 | ACCT 3@, ACCT 102, ACCT 1@, ACCT 2@ |
| 3 | Altuve, Jose | RA000456 | AAS Degree Requirements | 39 | 62.947369 | |
| 4 | Altuve, Jose | RA000736 | Major in Accounting (2021) | 33 | 37.25 | MATH 121, ACCT 110, MGMT 105, MGMT 106, MGMT 241, BUS 1@ |

# CPA Data Overview

- Must run extra process (DAP25) to create the CPA data
- Data gets created in **DAP_RESULT_DTL** and DAP_RESCLASS_DTL
- Data gets overwritten each time the CPA process runs (DAP25)
- DAP_RESULT_DTL can get BIG!
- Recommend turning on the flag (set to TRUE) **core.audit.cpa.latestTermOnly**

# Exceptions Reporting

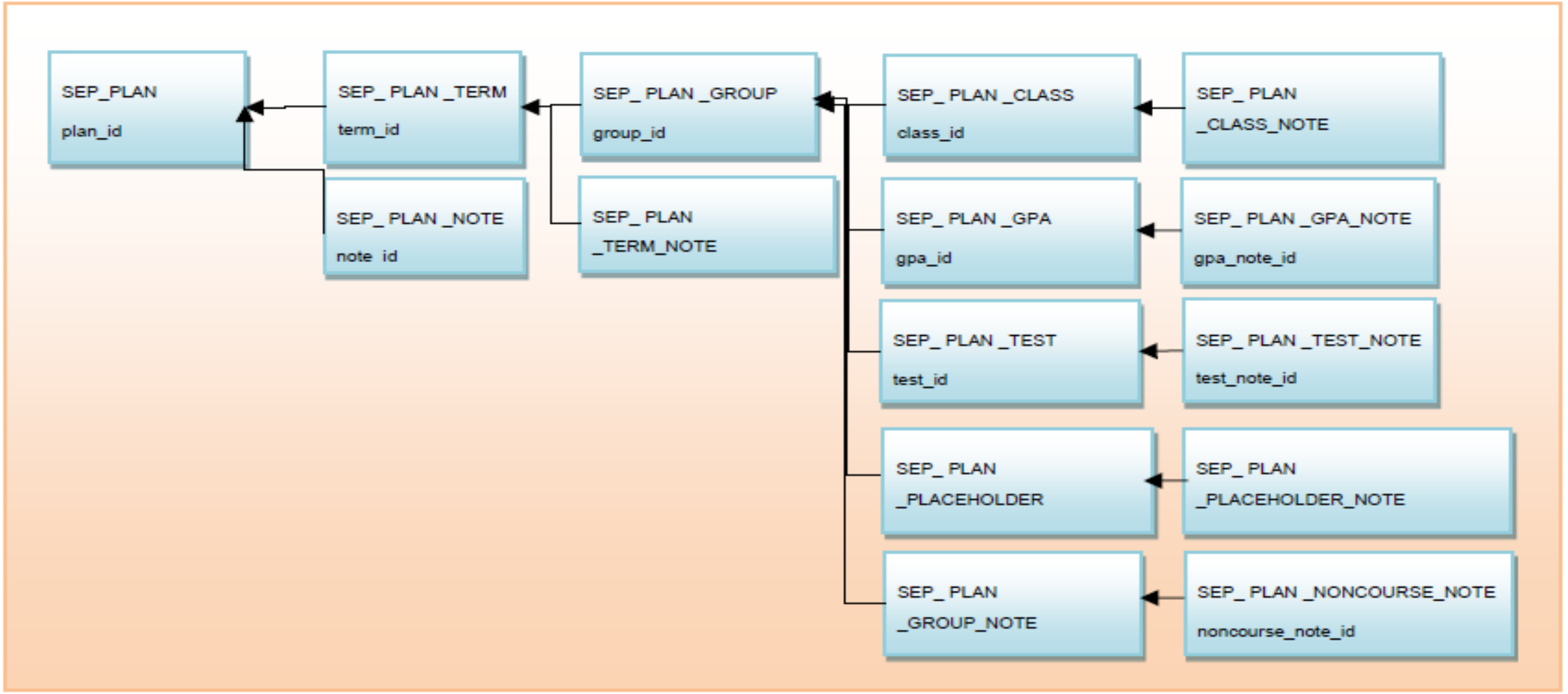
- Exceptions are stored in both the **DAP_EXCEPT_DTL** and the CPA tables
- There are some “canned” reports under the Exception Management tab or can write your own
- DAP_EXCEPT_DTL
  - DAP_EXC_TYPE – contains the type of exception that was entered
  - DAP_EXC_BUFFER – depending on the exception type contains information about the exception

# Student Educational Planner (SEP)

- SEP Tables are similar to Banner tables
- Created as soon as the student's plan or template is created
- Updated as plans and/or templates are changed
- No external process needed
- Highly normalized, so must do several joins to get down to course information

# SEP Table Structure

## Plan Tables



# SEP_PLAN

- **SEP_PLAN** is the top-level table
- Only table in the SEP tables that contains the student's ID
- One record per student plan
- Contains:
  - Description of the plan
  - Locked status
  - Active status
  - Tracking status
  - Plan creator and modifier

# SEP_PLAN_TERM

- Contains the term information for the plan
- Links back to SEP_PLAN with the PLAN_ID
- One record per term per plan
- Contains
  - Term Information
  - Term Tracking status

sep_plan.plan_id = sep_plan_term.plan_id

# SEP_PLAN_GROUP

- One record per plan
- Ties the term to the lower-level items
  - Other groups
  - Courses
  - Tests
  - Placeholders

  sep_plan_term.term_id = sep_plan_group.term_id

# SEP_PLAN_CLASS

- One record per course
- Ties the course to the group, term, plan
- Contains
  - Course discipline and number
  - Course attribute
  - Critical status
  - Tracking status

sep_plan_group.group_id = sep_plan_class.group_id

# Tying it all together

**To join student's plan together with classes in the plan**

```
select student_id, description, term, course_discipline,
course_number
from sep_plan , sep_plan_term , sep_plan_group , sep_plan_class
where sep_plan.is_active = 'Y'
and sep_plan.plan_id = sep_plan_term.plan_id
and sep_plan_term.group_id = sep_plan_group.group_id
and sep_plan_group.group_id = sep_plan_class.group_id
order by student_id, sep_plan.plan_id, sep_plan_term.term;
```

# List of classes planned

```
select student_id as "Student"
, description as "Plan Description"
, term as "Planned Term"
, RTRIM(course_discipline)||' '||RTRIM(course_number) as "Planned Course"
from sep_plan
, sep_plan_term
, sep_plan_group
, sep_plan_class
where sep_plan.is_active = 'Y'
and sep_plan.plan_id = sep_plan_term.plan_id
and sep_plan_term.group_id = sep_plan_group.group_id
and sep_plan_group.group_id = sep_plan_class.group_id
order by student_id
, sep_plan.plan_id
, sep_plan_term.term;
```

# List of classes planned

| | Student | Plan Description | Planned Term | Planned Course |
|---|---|---|---|---|
| 1 | N00047785 | AUGUST 2022 PLAN | 202208 | ACCT 104 |
| 2 | N00047785 | AUGUST 2022 PLAN | 202208 | BUS 102 |
| 3 | N00047785 | AUGUST 2022 PLAN | 202208 | BUS 100 |
| 4 | N00047785 | AUGUST 2022 PLAN | 202208 | ACCT 111 |
| 5 | N00047875 | BBA Fall 2022 | 202280 | ACCT 107 |
| 6 | N00047875 | BBA Fall 2022 | 202280 | COMM 104 |
| 7 | N00047875 | BBA Fall 2022 | 202280 | ECON 103 |
| 8 | N00047875 | BBA Fall 2022 | 202310 | BIOL 100 |
| 9 | N00047875 | BBA Fall 2022 | 202310 | MATH 100 |
| 10 | N00047875 | BBA Fall 2022 | 202310 | ACCT 200 |
| 11 | N00047875 | BBA Fall 2022 | 202310 | ACCT 301 |
| 12 | N00047875 | BBA Fall 2022 | 202310 | MATH 117 |
| 13 | N00047875 | BBA Fall 2022 | 202310 | ARH 2000 |
| 14 | N00047875 | BBA Fall 2022 | 202123 | ECON 112 |
| 15 | N00047875 | BBA Fall 2022 | 202123 | ECON 202 |
| 16 | N00047875 | BBA Fall 2022 | 202123 | ECON 203 |
| 17 | N00047875 | BBA Fall 2022 | 202280 | ACCT 107 |
| 18 | N00047875 | BBA Fall 2022 | 202280 | ACCT 301 |
| 19 | N00047875 | BBA Fall 2022 | 202280 | ECON 103 |

# Count of when and how many times courses are planned

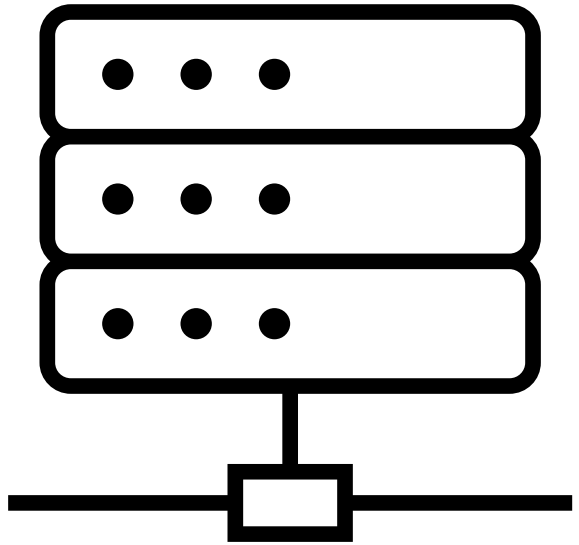
```
select term||' '||course_discipline||' '||course_number as "Term - Class"
, count(term||' '||course_discipline||' '||course_number) as "Times Planned"
from sep_plan
, sep_plan_term
, sep_plan_group
, sep_plan_class
where sep_plan.is_active = 'Y'
and sep_plan.plan_id = sep_plan_term.plan_id
and sep_plan_term.group_id = sep_plan_group.group_id
and sep_plan_group.group_id = sep_plan_class.group_id
group by (term||' '||course_discipline||' '||course_number)
order by (term||' '||course_discipline||' '||course_number);
```

# Count of when and how many times courses are planned

| | Term - Class | Times Planned |
|---|---|---|
| 1 | 202010 POLS 101 | 1 |
| 2 | 202123 ECON 112 | 1 |
| 3 | 202123 ECON 202 | 1 |
| 4 | 202123 ECON 203 | 1 |
| 5 | 202208 ACCT 104 | 1 |
| 6 | 202208 ACCT 111 | 1 |
| 7 | 202208 BUS 100 | 1 |
| 8 | 202208 BUS 102 | 1 |
| 9 | 202210 ACCT 200 | 1 |
| 10 | 202220 0119 001 | 1 |
| 11 | 202220 ART 100 | 1 |
| 12 | 202280 ACCT 107 | 2 |
| 13 | 202280 ACCT 110 | 1 |
| 14 | 202280 ACCT 301 | 1 |
| 15 | 202280 COMM 104 | 2 |
| 16 | 202280 COMM 110 | 1 |
| 17 | 202280 ECON 103 | 2 |
| 18 | 202280 INDI 110 | 1 |

# Degree Works Reporting

- Degree Works contains a great deal of data about student audits and plans
- Hopefully now you can have some ideas on how to retrieve it!



SIGCONNECT
Higher Education Conference


QUESTIONS
SIGCONNECT
Higher Education Conference

# Contact


## Dennis Crowson

Senior Consultant – Degree Works

crowson@sigcorp.com


/strata-information-group

@SIGCorpLIVE

Sigcorp.com


THANK
YOU
SIGCONNECT
Higher Education Conference