PTDB DBSchema Documentation

Layouts

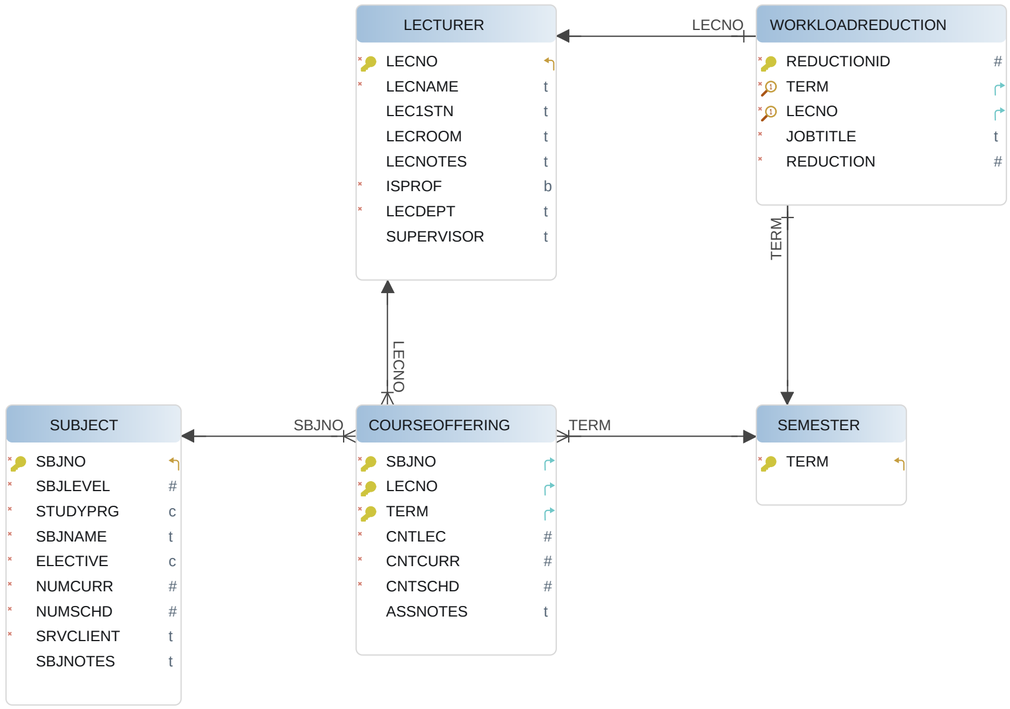
1. Default Diagram 1

Tables

DB2INST1.COURSEOFFERING [1] 2
DB2INST1.LECTURER [1] 2
DB2INST1.SEMESTER [1] 2
DB2INST1.SUBJECT [1] 3
DB2INST1.WORKLOADREDUCTION [1] 3

Views

DB2INST1.ALERTFORMISSINGCOURSES [1] 4
DB2INST1.EXPANSIVELECTURES [1] 4
DB2INST1.EXPORTEDSUBJECTS [1] 4
DB2INST1.IMPORTLECTURERS [1] 4
DB2INST1.LECTURERSWITHLOAD [1] 4
DB2INST1.PROFESSORSWITHREDUCTION [1] 4
DB2INST1.PROFESSORWORKLOAD [1] 5
DB2INST1.STAFFMEMBERS [1] 5
DB2INST1.TEACHINGPROFESSORS [1] 5
DB2INST1.TERMLIST [1] 5
DB2INST1.WORKLOADBALANCE [1] 5



TERMLIST

Counts all terms in database

STAFFMEMBERS

Differentiates between professors and contract teachers

LECTURERSWITHLOAD

Lists lecturers with 2 or 3 courses per term

EXPANSIVELECTURES

Lists all the lectures which have more workload than hours given by the curriculum

TEACHINGPROFESSORS

PROFESSORSWITHREDUCTION

Lists all professors who are currently teaching

Lists professors with additional workload (positive or negative)

PROFESSORWORKLOAD

Shows the workload of each professor

WORKLOADBALANCE

LECNO	#
TERM	
TEACH_HOURS	#
REDUCTION_HOURS	#
TOTAL_WORKLOAD	#

Shows the workload through out the terms including the reductions

IMPORTLECTURERS

Listing lecturers teaching a subject with different department

EXPORTEDSUBJECTS

Listing subjects taught by lecturers with different departments

ALERTFORMISSINGCOURSES

SBJNO	t
SBJNAME	t

All subjects, that are not taught can be found through this view. So it is clear which courses have to be planned.

Default Diagram

Table COURSEOFFERING

Idx	Name	Data Type
* Pk	SBJNO	VARCHAR(25)
* Pk	LECNO	INTEGER
* Pk	TERM	VARCHAR(20)
*	CNTLEC	DECIMAL(3,2)
*	CNTCURR	INTEGER
*	CNTSCHD	INTEGER
	ASSNOTES	VARCHAR(150)

Indexes

Type	Name	On
Pk	PK_COURSEOFFERING	SBJNO, LECNO, TERM

Foreign Keys

Type	Name	On
	FK_COURSEOFFERING_SUBJECT (SBJNO) ref SUBJECT (SBJNO)	
	FK_COURSEOFFERING_LECTURER (LECNO) ref LECTURER (LECNO)	
	FK_COURSEOFFERING_SEMESTER (TERM) ref SEMESTER (TERM)	

Table LECTURER

Idx	Name	Data Type
* Pk	LECNO	INTEGER
*	LECNAME	VARCHAR(100)
	LEC1STN	VARCHAR(100)
	LECROOM	VARCHAR(20)
	LECNOTES	VARCHAR(150)
*	ISPROF	BOOLEAN
*	LECDEPT	VARCHAR(5)
	SUPERVISOR	VARCHAR(100)

Indexes

Type	Name	On
Pk	PK_LECTURER	LECNO

Table SEMESTER

Idx	Name	Data Type
* Pk	TERM	VARCHAR(20)

Table SEMESTER

Indexes

Type	Name	On
Pk	PK_SEMESTER	TERM

Table SUBJECT

Idx	Name	Data Type
* Pk	SBJNO	VARCHAR(25)
*	SBJLEVEL	INTEGER
*	STUDYPRG	CHAR(10)
*	SBJNAME	VARCHAR(100)
*	ELECTIVE	CHAR(10)
*	NUMCURR	INTEGER
*	NUMSCHD	INTEGER
*	SRVCLIENT	VARCHAR(10)
	SBJNOTES	VARCHAR(150)

Indexes

Type	Name	On
Pk	PK_SUBJECT	SBJNO

Table WORKLOADREDUCTION

Idx	Name	Data Type
* Pk	REDUCTIONID	INTEGER GENERATED ALWAYS AS IDENTITY (START WITH 1 INCREMENT BY 1)
* Unq	TERM	VARCHAR(20)
* Unq	LECNO	INTEGER
*	JOBTITLE	VARCHAR(100)
*	REDUCTION	INTEGER

Indexes

Type	Name	On
Pk	PK_WORKLOADREDUCTION	REDUCTIONID
Unq	UNQ_LEC_NO	LECNO
Unq	UNQ_TERM	TERM

Foreign Keys

Type	Name	On
	FK_WORKLOADREDUCTION_LECTURER (LECNO) ref LECTURER (LECNO)	
	FK_WORKLOADREDUCTION_SEMESTER (TERM) ref SEMESTER (TERM)	

View ALERTFORMISSINGCOURSES

```
CREATE OR REPLACE VIEW ${nameWithSchemaName} AS
SELECT S.SBJNO, S.SBJNAME
FROM DB2INST1.SUBJECT S
LEFT OUTER JOIN COURSEOFFERING C ON S.SBJNO = C.SBJNO
WHERE C.SBJNO IS NULL;
```

View EXPANSIVELECTURES

```
CREATE VIEW ${nameWithSchemaName} AS
SELECT SBJNO, SBJNAME, NUMSCHD, NUMCURR
FROM DB2INST1.SUBJECT
WHERE NUMSCHD > NUMCURR;
```

View EXPORTEDSUBJECTS

```
CREATE VIEW ${nameWithSchemaName} AS
SELECT *
FROM (
    SELECT L.LECDEPT AS department, COUNT(*) AS export_count
    FROM DB2INST1.LECTURER as L
    JOIN DB2INST1.COURSEOFFERING as C on L.LECNO = C.LECNO
    JOIN DB2INST1.SUBJECT as S on C.SBJNO = S.SBJNO
    WHERE L.LECDEPT != S.SRVCLIENT
    GROUP BY L.LECDEPT
    ORDER BY L.LECDEPT
) as SUB;
```

View IMPORTLECTURERS

```
CREATE VIEW ${nameWithSchemaName} AS
SELECT *
FROM (
    SELECT S.SRVCLIENT AS department, COUNT(*) AS import_count
    FROM DB2INST1.SUBJECT as S
    JOIN DB2INST1.COURSEOFFERING as C on C.SBJNO = S.SBJNO
    JOIN DB2INST1.LECTURER as L on L.LECNO = C.LECNO
    WHERE S.SRVCLIENT != L.LECDEPT
    GROUP BY S.SRVCLIENT
    ORDER BY S.SRVCLIENT
) as SUB;
```

View LECTURERSWITHLOAD

```
CREATE VIEW ${nameWithSchemaName} AS
SELECT LECNO, TERM, COUNT(*) AS offerings
FROM DB2INST1.COURSEOFFERING
GROUP BY LECNO, TERM
HAVING COUNT(*) IN (2,3);
```

View PROFESSORSWITHREDUCTION

```
CREATE VIEW ${nameWithSchemaName} AS
SELECT DISTINCT L.LECNO
FROM DB2INST1.LECTURER AS L
JOIN DB2INST1.WORKLOADREDUCTION AS R ON R.LECNO = L.LECNO
WHERE L.ISPROF = TRUE;
```

View PROFESSORWORKLOAD

```
CREATE VIEW ${nameWithSchemaName} AS
SELECT LECNO, TERM, SUM(CNTLEC) AS total_lec_hrs
FROM DB2INST1.COURSEOFFERING
GROUP BY LECNO, TERM;
```

View STAFFMEMBERS

```
CREATE VIEW ${nameWithSchemaName} AS
SELECT ISPROF, COUNT(*) AS cnt
FROM DB2INST1.LECTURER
GROUP BY ISPROF;
```

View TEACHINGPROFESSORS

```
CREATE VIEW ${nameWithSchemaName} AS
SELECT DISTINCT L.LECNO
FROM DB2INST1.LECTURER AS L
JOIN DB2INST1.COURSEOFFERING AS C ON C.LECNO = L.LECNO
WHERE L.ISPROF = TRUE
AND NOT EXISTS (
    SELECT 1
    FROM DB2INST1.WORKLOADREDUCTION AS R
    WHERE R.LECNO = L.LECNO
);
```

View TERMLIST

```
CREATE VIEW ${nameWithSchemaName} AS
SELECT COUNT (DISTINCT term) AS term_count
FROM COURSEOFFERING G;
```

View WORKLOADBALANCE

```
CREATE VIEW ${nameWithSchemaName} AS
WITH
  teach AS (
    SELECT
      C.lecno,
      C.term,
      SUM(
        CASE
          WHEN C.assNotes = 'gekoppelt mit WKB3' THEN 0
          ELSE C.CNTLEC
        END
      ) AS teach_hours
    FROM DB2INST1.CourseOffering AS C
    GROUP BY C.lecno, C.term
  ),
  red AS (
    SELECT
      R.lecno,
      R.term,
      SUM(R.reduction) AS reduction_hours
    FROM DB2INST1.WorkloadReduction AS R
    GROUP BY R.lecno, R.term
  )
SELECT
  t.lecno,
  t.term,
  t.teach_hours,
  COALESCE(r.reduction_hours,0)           AS reduction_hours,
  t.teach_hours
    + COALESCE(r.reduction_hours,0)       AS total_workload
FROM teach AS t
LEFT JOIN red   AS r
  ON t.lecno = r.lecno
 AND t.term  = r.term
;
```