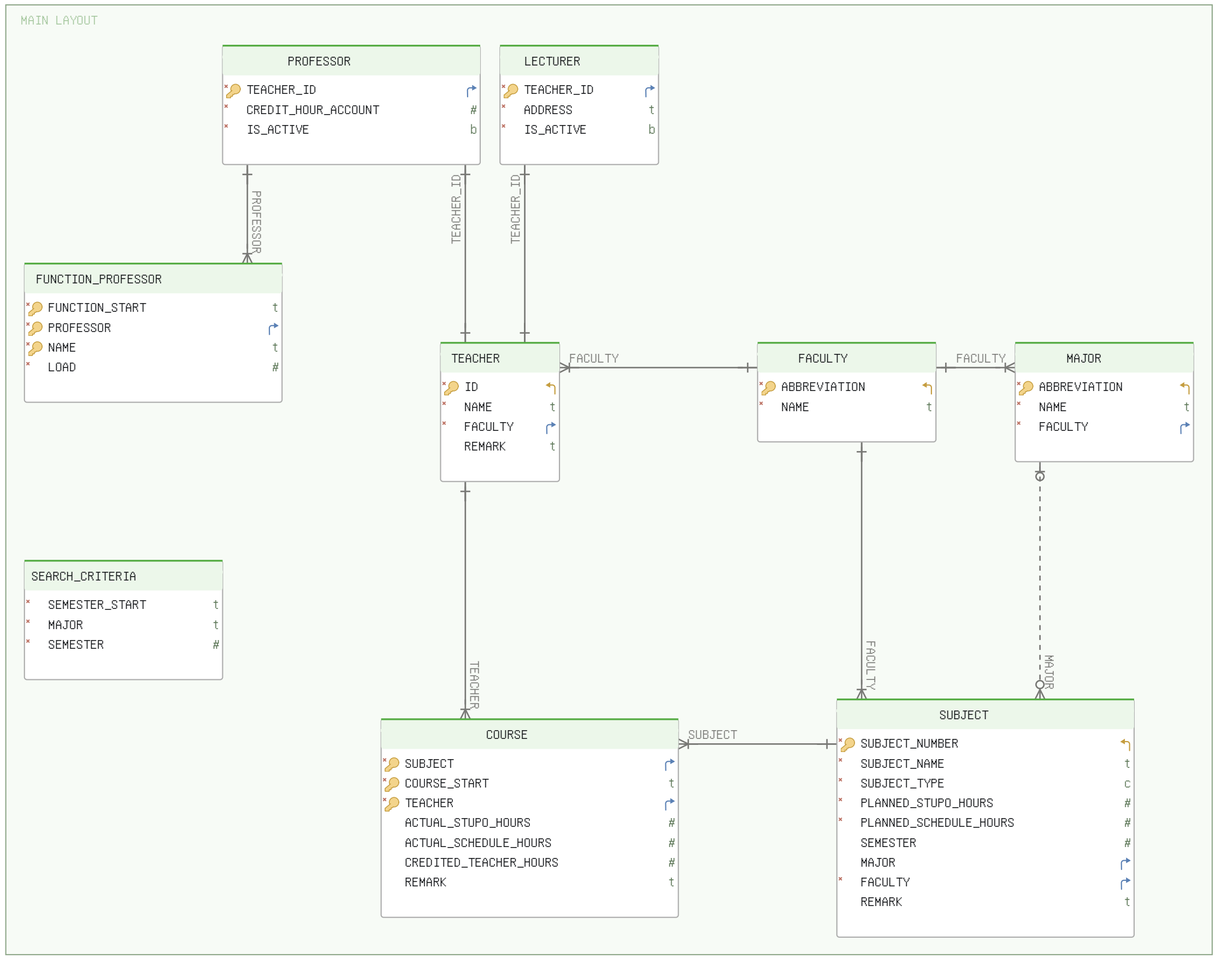
Layouts

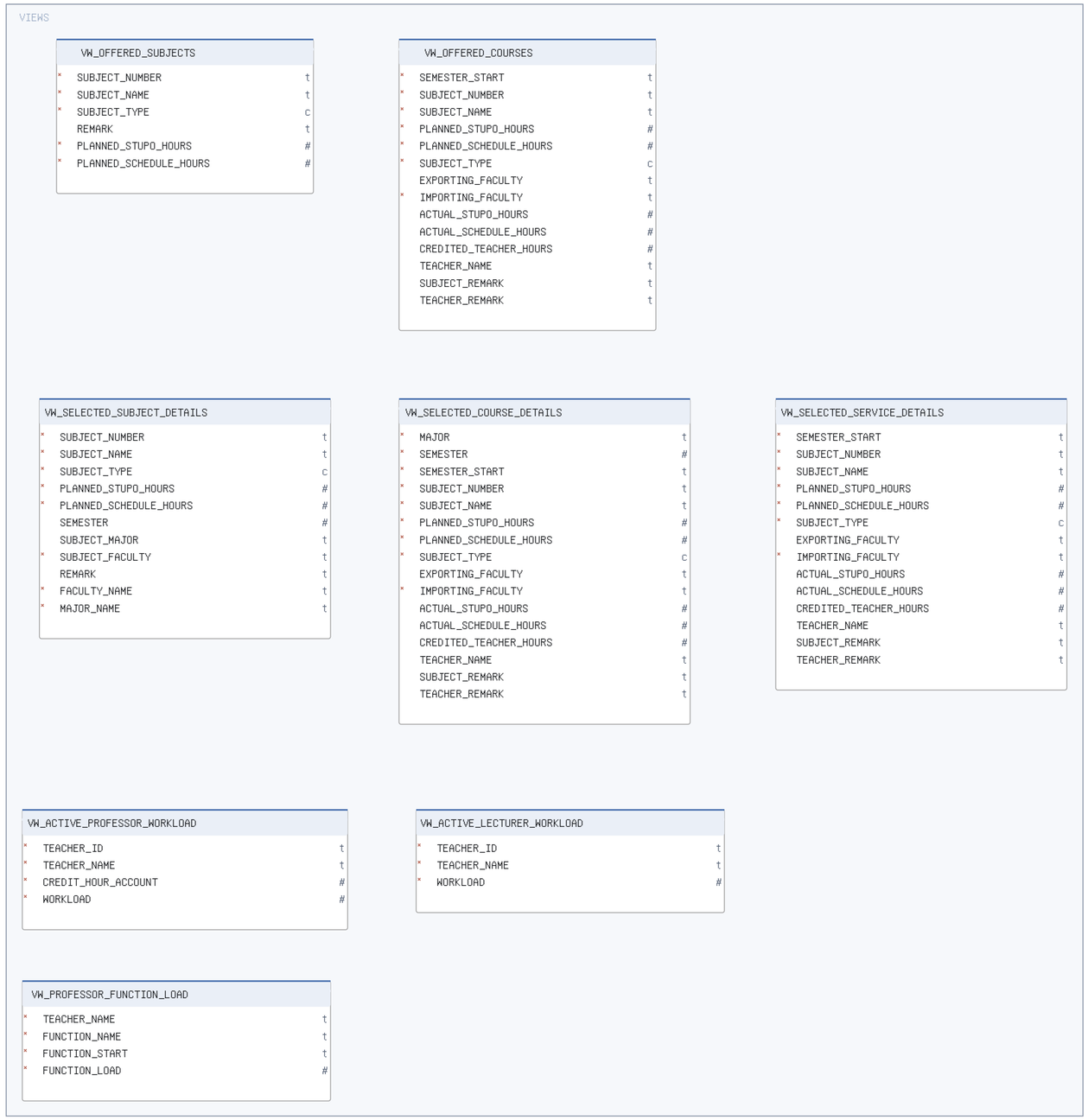
Tables

Views

MAIN LAYOUT



VIEWS



Main Layout

Table COURSE

Idx	Name	Data Type	Description
A class that is held by a teacher in a specific semester (such as "WS24/25").			
* Pk	SUBJECT	VARCHAR(15)	The subject (number) which is taught in this course. E.g. "R39.04995"
* Pk	COURSE_START	VARCHAR(10)	The semester in which a class starts. E.g. "WS24/25"
* Pk	TEACHER	VARCHAR(15)	The teacher (id) teaching this course.
	ACTUAL_STUPO_HOURS	INTEGER	The actual hours for this subject according to the StuPo. E.g. "2"
	ACTUAL_SCHEDULE_HOURS	INTEGER	The actual hours for this subject according to the schedule. E.g. "2"
	CREDITED_TEACHER_HOURS	INTEGER	The hours that will be credited to the teacher. This is part of the teacher's workload. E.g. "1"
	REMARK	VARCHAR(255)	An optional remark about the course.

Indexes

Type	Name	On	Description
Pk	PK_COURSE	SUBJECT, COURSE_START, TEACHER	

Foreign Keys

Type	Name	On	Description
	FK_SUBJECT_TEACHER_SUBJECT (SUBJECT) ref SUBJECT (SUBJECT_NUMBER)		
	FK_SUBJECT_TEACHER_TEACHER (TEACHER) ref TEACHER (ID)		

Constraints

	Name	Definition	Description
	CNS_COURSE	ACTUAL_SCHEDULE_HOURS % 2 = 0 AND ACTUAL_SCHEDULE_HOURS >= 0 AND ACTUAL_SCHEDULE_HOURS <= 10	
	CNS_COURSE_0	ACTUAL_STUPO_HOURS >= 0 AND ACTUAL_STUPO_HOURS <= 10	
	CNS_COURSE_1	CREDITED_TEACHER_HOURS >= 0 AND CREDITED_TEACHER_HOURS <= 10	

Triggers

	Name	Definition	Description
	TR_INSERT_COURSE		

Table COURSE

```
CREATE OR REPLACE TRIGGER ${nameWithSchemaName}
BEFORE INSERT ON INSTANCE.COURSE
REFERENCING NEW AS NEW_ROW
FOR EACH ROW
BEGIN
    DECLARE is_active BOOLEAN DEFAULT FALSE;
    -- is prof active?
    SELECT IS_ACTIVE INTO is_active
    FROM INSTANCE.PROFESSOR
    WHERE TEACHER_ID = NEW_ROW.TEACHER;
    -- if prof inactive / not found, check the lecturer table
    IF is_active = FALSE THEN
        SELECT IS_ACTIVE INTO is_active
        FROM INSTANCE.LECTURER
        WHERE TEACHER_ID = NEW_ROW.TEACHER;
    END IF;
    --if inactive for both
    IF is_active = FALSE THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Cannot insert course for inactive professor or lecturer';
    END IF;
END
```

Table FACULTY

Idx	Name	Data Type	Description
An academic unit that specializes in a particular field and contains related subjects			
* Pk	ABBREVIATION	VARCHAR(5)	An identifying abbreviation for the faculty. E.g. "IT"
*	NAME	VARCHAR(100)	The full name of the faculty. E.g. "Information Technology"

Indexes

Type	Name	On	Description
Pk	PK_FACULTY	ABBREVIATION	

Table FUNCTION_PROFESSOR

Idx	Name	Data Type	Description
Responsibilities undertaken by a professor related to the management and organization of academic programs, faculty governance, and departmental operations.			
* Pk	FUNCTION_START	VARCHAR(10)	The semester in which a function starts. E.g. "WS24/25"
* Pk	PROFESSOR	VARCHAR(15)	The professor (id) which exercises this function.
* Pk	NAME	VARCHAR(100)	The name of the function. E.g. "Head of internship office"
*	LOAD	INTEGER	The hourly load of the function. E.g. "2"

Indexes

Type	Name	On	Description
Pk	PK_FUNC_PROF_NAME	FUNCTION_START, PROFESSOR, NAME	

Foreign Keys

Type	Name	On	Description
	FK_FUNCTION_PROFESSOR (PROFESSOR) ref PROFESSOR (TEACHER_ID)		

Constraints

	Name	Definition	Description
	CNS_FUNCTION	LOAD >= 0 AND LOAD <= 10	

Table LECTURER

Idx	Name	Data Type	Description
A contracted teacher, also known as "lecturer" in the project document.			
* Pk	TEACHER_ID	VARCHAR(15)	An identifying number for the teacher. Foreign Key from the teacher table.

Table LECTURER

*	ADDRESS	VARCHAR(100)	Address of the lecturer. In order to be able to write to the lecturers. E.g. "Kanalstraße 33, Esslingen"
*	IS_ACTIVE	BOOLEAN	

Indexes

Type	Name	On	Description
Pk	PK_LECTURER	TEACHER_ID	

Foreign Keys

Type	Name	On	Description
	FK_LECTURER_TEACHER (TEACHER_ID) ref TEACHER (ID)		

Triggers

	Name	Definition	Description
	TR_CHECK_EXISTING_PROFESSOR		

```
CREATE OR REPLACE TRIGGER ${nameWithSchemaName}
BEFORE INSERT ON instance.lecturer
REFERENCING NEW AS NEW_ROW
FOR EACH ROW
BEGIN
    IF EXISTS (SELECT 1 FROM instance.professor WHERE teacher_id = NEW_ROW.teacher_id) THEN
        SIGNAL SQLSTATE '45000';
    END IF;
END
```

Table MAJOR

Idx	Name	Data Type	Description
A degree program.			
* Pk	ABBREVIATION	VARCHAR(5)	An identifying abbreviation for the major. E.g. "SWB"
*	NAME	VARCHAR(100)	The full name for the major. E.g. "Software Engineering and Media Computing"
*	FACULTY	VARCHAR(5)	The faculty a major belongs to. E.g. "IT" for "Software Engineering and Media Computing"

Indexes

Type	Name	On	Description
Pk	PK_MAJOR	ABBREVIATION	

Foreign Keys

Type	Name	On	Description
	FK_MAJOR_FACULTY (FACULTY) ref FACULTY (ABBREVIATION)		

Table PROFESSOR

Idx	Name	Data Type	Description
A permanently employed academic who is responsible for teaching and research, also known as "professor" in the project document.			
* Pk	TEACHER_ID	VARCHAR(15)	An identifying number for the professor. Foreign Key from the teacher table.
*	CREDIT_HOUR_ACCOUNT	DECIMAL(5,0) DEFAULT 0	Additional work beyond the 18 compulsory work hours for a professor is credited to a credit hour account, less work is debited from this account. E.g. "2"
*	IS_ACTIVE	BOOLEAN	

Indexes

Type	Name	On	Description
Pk	PK_PROFESSOR	TEACHER_ID	

Foreign Keys

Type	Name	On	Description
	FK_PROFESSOR_TEACHER (TEACHER_ID) ref TEACHER (ID)		

Triggers

Table PROFESSOR

	Name	Definition	Description
	TR_CHECK_EXISTING_LECTURER		

```
CREATE OR REPLACE TRIGGER ${nameWithSchemaName}
BEFORE INSERT ON instance.professor
REFERENCING NEW AS NEW_ROW
FOR EACH ROW
BEGIN
    IF EXISTS (SELECT 1 FROM instance.lecturer WHERE teacher_id = NEW_ROW.teacher_id) THEN
        SIGNAL SQLSTATE '45000';
    END IF;
END
```

Table SEARCH_CRITERIA

Idx	Name	Data Type	Description
A table necessary for user views.			
*	SEMESTER_START	VARCHAR(10)	The semester for which the data is shown. E.g. "WS24/25"
*	MAJOR	VARCHAR(5)	The major for which the data is shown. E.g. "SWB"
*	SEMESTER	INTEGER	The semester of a major for which the data is shown. E.g. "6" for "SWB"

Triggers

	Name	Definition	Description
	TR_UPDATE_CREDIT_HOUR_ACCOUNT		

```
CREATE OR REPLACE TRIGGER ${nameWithSchemaName}
AFTER UPDATE OF semester_start ON INSTANCE.SEARCH_CRITERIA
REFERENCING OLD AS OLD_ROW NEW AS NEW_ROW
FOR EACH ROW
BEGIN
    -- Only if there is a difference
    IF OLD_ROW.semester_start <> NEW_ROW.semester_start THEN
        -- Call stored procedure with the new semester_start value
        CALL INSTANCE.SP_UPDATE_CREDIT_HOUR_ACCOUNT(NEW_ROW.semester_start);
    END IF;
END
```

Table SUBJECT

Idx	Name	Data Type	Description
A specific subject listed in the curriculum of a major.			
* Pk	SUBJECT_NUMBER	VARCHAR(15)	An identifying number for the subject. E.g. "R39.04995"
*	SUBJECT_NAME	VARCHAR(100)	The name of a subject. E.g. "Information Systems"
*	SUBJECT_TYPE	CHAR(1) DEFAULT 'P'	If the subject is an compulsory "P", elective "W" or supplementary "Z" E.g. "P" for "Information Systems"
*	PLANNED_STUPO_HOURS	INTEGER	The planned hours for this subject according to the StuPo. E.g. "2"
*	PLANNED_SCHEDULE_HOURS	INTEGER	The planned hours for this subject according to the schedule. E.g. "2"
	SEMESTER	INTEGER	The planned semester for this subject according to the curriculum. E.g. "6" for "Information Systems"
	MAJOR	VARCHAR(5)	The major to which this subject belongs to. E.g. "SWB"
*	FACULTY	VARCHAR(5)	The faculty of this subject. E.g. "IT"
	REMARK	VARCHAR(255)	An optional remark about the subject.

Indexes

Type	Name	On	Description
Pk	PK_SUBJECT	SUBJECT_NUMBER	

Foreign Keys

Type	Name	On	Description
	FK_SUBJECT_FACULTY (FACULTY) ref FACULTY (ABBREVIATION)		
	FK_SUBJECT_MAJOR (MAJOR) ref MAJOR (ABBREVIATION)		

Table SUBJECT

Constraints

	Name	Definition	Description
	CNS_SUBJECT	SEMESTER BETWEEN 1 AND 7	
	PSCHH_SUBJECT	PLANNED_SCHEDULE_HOURS >= 0 AND PLANNED_SCHEDULE_HOURS <= 10	
	PSTUH_SUBJECT	PLANNED_STUPO_HOURS >= 0 AND PLANNED_STUPO_HOURS <= 10	

Table TEACHER

Idx	Name	Data Type	Description
A person teaching a course.			
* Pk	ID	VARCHAR(15)	An identifying number for the teacher.
*	NAME	VARCHAR(100)	The name of the teacher. E.g. "Steffen Schober"
*	FACULTY	VARCHAR(5)	The faculty the teacher belongs to. E.g. "IT" for "Steffen Schober"
	REMARK	VARCHAR(255)	An optional remark made by the teacher about themselves.

Indexes

Type	Name	On	Description
Pk	PK_TEACHER	ID	

Foreign Keys

Type	Name	On	Description
	FK_TEACHER_FACULTY (FACULTY) ref FACULTY (ABBREVIATION)		

View VW_ACTIVE_LECTURER_WORKLOAD

```
CREATE OR REPLACE VIEW ${nameWithSchemaName} AS SELECT t.ID AS TEACHER_ID, t.NAME AS TEACHER_NAME, COALESCE(SUM(c.CREDITED_TEACHER_HOURS), 0) AS WORKLOAD FROM INSTANCE.LECTURER l JOIN INSTANCE.TEACHER t ON t.ID = l.TEACHER_ID LEFT JOIN INSTANCE.COURSE c ON c.TEACHER = t.ID JOIN INSTANCE.SEARCH_CRITERIA sc ON sc.SEMESTER_START = c.COURSE_START WHERE l.IS_ACTIVE = TRUE GROUP BY t.ID, t.NAME
```

View VW_ACTIVE_PROFESSOR_WORKLOAD

```
CREATE OR REPLACE VIEW ${nameWithSchemaName} AS SELECT t.ID AS TEACHER_ID, t.NAME AS TEACHER_NAME, p.CREDIT_HOUR_ACCOUNT, COALESCE(SUM(c.CREDITED_TEACHER_HOURS), 0) AS WORKLOAD FROM INSTANCE.PROFESSOR p JOIN INSTANCE.TEACHER t ON t.ID = p.TEACHER_ID LEFT JOIN INSTANCE.COURSE c ON c.TEACHER = t.ID JOIN INSTANCE.SEARCH_CRITERIA sc ON sc.SEMESTER_START = c.COURSE_START WHERE p.IS_ACTIVE = TRUE GROUP BY t.ID, t.NAME, p.CREDIT_HOUR_ACCOUNT
```

View VW_OFFERED_COURSES

```
CREATE OR REPLACE VIEW ${nameWithSchemaName} AS SELECT sc.SEMESTER_START, s.SUBJECT_NUMBER, s.SUBJECT_NAME, s.PLANNED_STUPO_HOURS, s.PLANNED_SCHEDULE_HOURS, s.SUBJECT_TYPE, t.FACULTY AS EXPORTING_FACULTY, s.FACULTY AS IMPORTING_FACULTY, c.ACTUAL_STUPO_HOURS, c.ACTUAL_SCHEDULE_HOURS, c.CREDITED_TEACHER_HOURS, t.NAME AS TEACHER_NAME, s.REMARK AS SUBJECT_REMARK, t.REMARK AS TEACHER_REMARK FROM INSTANCE.SUBJECT s LEFT JOIN INSTANCE.COURSE c ON c.SUBJECT = s.SUBJECT_NUMBER JOIN INSTANCE.SEARCH_CRITERIA sc ON c.COURSE_START = sc.SEMESTER_START LEFT JOIN INSTANCE.TEACHER t ON t.ID = c.TEACHER
```

View VW_OFFERED_SUBJECTS

```
CREATE OR REPLACE VIEW ${nameWithSchemaName} AS SELECT su.SUBJECT_NUMBER, su.SUBJECT_NAME, su.SUBJECT_TYPE, su.REMARK, su.PLANNED_STUPO_HOURS, su.PLANNED_SCHEDULE_HOURS FROM INSTANCE.SUBJECT su
```

View VW_PROFESSOR_FUNCTION_LOAD

```
CREATE OR REPLACE VIEW ${nameWithSchemaName} AS SELECT t.NAME AS TEACHER_NAME, f.NAME AS FUNCTION_NAME, f.FUNCTION_START, f.LOAD AS FUNCTION_LOAD FROM INSTANCE.FUNCTION_PROFESSOR f JOIN INSTANCE.PROFESSOR p ON p.TEACHER_ID = f.PROFESSOR JOIN INSTANCE.TEACHER t ON t.ID = p.TEACHER_ID JOIN INSTANCE.SEARCH_CRITERIA sc ON f.FUNCTION_START = sc.SEMESTER_START WHERE p.IS_ACTIVE = TRUE
```

View VW_SELECTED_COURSE_DETAILS

```
CREATE OR REPLACE VIEW ${nameWithSchemaName} AS SELECT sc.MAJOR, sc.SEMESTER, sc.SEMESTER_START, s.SUBJECT_NUMBER, s.SUBJECT_NAME, s.PLANNED_STUPO_HOURS, s.PLANNED_SCHEDULE_HOURS, s.SUBJECT_TYPE, t.FACULTY AS EXPORTING_FACULTY, s.FACULTY AS IMPORTING_FACULTY, c.ACTUAL_STUPO_HOURS, c.ACTUAL_SCHEDULE_HOURS, c.CREDITED_TEACHER_HOURS, t.NAME AS TEACHER_NAME, s.REMARK AS SUBJECT_REMARK, t.REMARK AS TEACHER_REMARK FROM INSTANCE.SEARCH_CRITERIA sc JOIN INSTANCE.SUBJECT s ON s.MAJOR = sc.MAJOR AND s.SEMESTER = sc.SEMESTER LEFT JOIN INSTANCE.COURSE c ON c.SUBJECT = s.SUBJECT_NUMBER AND c.COURSE_START = sc.SEMESTER_START LEFT JOIN INSTANCE.TEACHER t ON t.ID = c.TEACHER
```

View VW_SELECTED_SERVICE_DETAILS

```
CREATE OR REPLACE VIEW ${nameWithSchemaName} AS SELECT sc.SEMESTER_START, s.SUBJECT_NUMBER, s.SUBJECT_NAME, s.PLANNED_STUPO_HOURS, s.PLANNED_SCHEDULE_HOURS, s.SUBJECT_TYPE, t.FACULTY AS EXPORTING_FACULTY, s.FACULTY AS IMPORTING_FACULTY, c.ACTUAL_STUPO_HOURS, c.ACTUAL_SCHEDULE_HOURS, c.CREDITED_TEACHER_HOURS, t.NAME AS TEACHER_NAME, s.REMARK AS SUBJECT_REMARK, t.REMARK AS TEACHER_REMARK FROM INSTANCE.SUBJECT s LEFT JOIN INSTANCE.COURSE c ON c.SUBJECT = s.SUBJECT_NUMBER JOIN INSTANCE.SEARCH_CRITERIA sc ON c.COURSE_START = sc.SEMESTER_START LEFT JOIN INSTANCE.TEACHER t ON t.ID = c.TEACHER WHERE t.FACULTY != s.FACULTY
```

View VW_SELECTED_SUBJECT_DETAILS

```
CREATE OR REPLACE VIEW ${nameWithSchemaName} AS SELECT su.SUBJECT_NUMBER, su.SUBJECT_NAME, su.SUBJECT_TYPE, su.PLANNED_STUPO_HOURS, su.PLANNED_SCHEDULE_HOURS, su.SEMESTER, su.MAJOR AS SUBJECT_MAJOR, su.FACULTY AS SUBJECT_FACULTY, su.REMARK, fa.NAME AS FACULTY_NAME, ma.NAME AS MAJOR_NAME FROM INSTANCE.SUBJECT su INNER JOIN INSTANCE.FACULTY fa ON fa.ABBREVIATION = su.FACULTY INNER JOIN INSTANCE.MAJOR ma ON ma.ABBREVIATION = su.MAJOR INNER JOIN INSTANCE.SEARCH_CRITERIA sc ON sc.MAJOR = su.MAJOR  -- Join mit SEARCH_CRITERIA
```

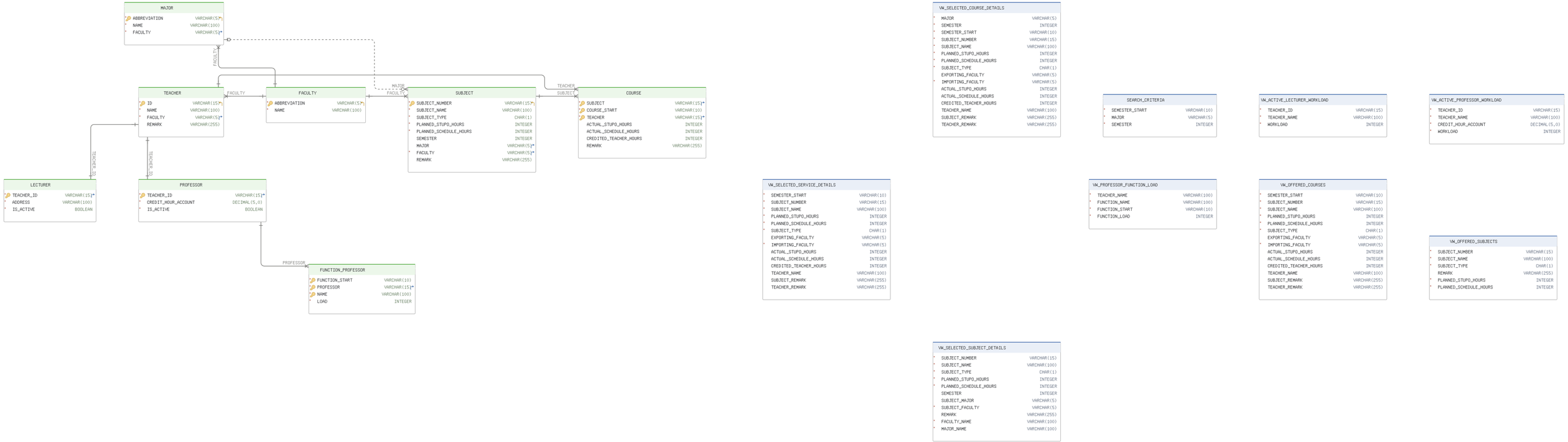
Schema INSTANCE

Functions

F_MISSING_COURSES_CHECK

Procedures

SP_UPDATE_CREDIT_HOUR_ACCOUNT



~Layout with Sample Tools

Table COURSE

Idx	Name	Data Type	Description
A class that is held by a teacher in a specific semester (such as "WS24/25").			
* Pk	SUBJECT	VARCHAR(15)	The subject (number) which is taught in this course. E.g. "R39.04995"
* Pk	COURSE_START	VARCHAR(10)	The semester in which a class starts. E.g. "WS24/25"
* Pk	TEACHER	VARCHAR(15)	The teacher (id) teaching this course.
	ACTUAL_STUPO_HOURS	INTEGER	The actual hours for this subject according to the StuPo. E.g. "2"
	ACTUAL_SCHEDULE_HOURS	INTEGER	The actual hours for this subject according to the schedule. E.g. "2"
	CREDITED_TEACHER_HOURS	INTEGER	The hours that will be credited to the teacher. This is part of the teacher's workload. E.g. "1"
	REMARK	VARCHAR(255)	An optional remark about the course.

Indexes

Type	Name	On	Description
Pk	PK_COURSE	SUBJECT, COURSE_START, TEACHER	

Foreign Keys

Type	Name	On	Description
	FK_SUBJECT_TEACHER_SUBJECT (SUBJECT) ref SUBJECT (SUBJECT_NUMBER)		
	FK_SUBJECT_TEACHER_TEACHER (TEACHER) ref TEACHER (ID)		

Constraints

	Name	Definition	Description
	CNS_COURSE	ACTUAL_SCHEDULE_HOURS % 2 = 0 AND ACTUAL_SCHEDULE_HOURS >= 0 AND ACTUAL_SCHEDULE_HOURS <= 10	
	CNS_COURSE_0	ACTUAL_STUPO_HOURS >= 0 AND ACTUAL_STUPO_HOURS <= 10	
	CNS_COURSE_1	CREDITED_TEACHER_HOURS >= 0 AND CREDITED_TEACHER_HOURS <= 10	

Triggers

	Name	Definition	Description
	TR_INSERT_COURSE		

Table COURSE

```
CREATE OR REPLACE TRIGGER ${nameWithSchemaName}
BEFORE INSERT ON INSTANCE.COURSE
REFERENCING NEW AS NEW_ROW
FOR EACH ROW
BEGIN
    DECLARE is_active BOOLEAN DEFAULT FALSE;
    -- is prof active?
    SELECT IS_ACTIVE INTO is_active
    FROM INSTANCE.PROFESSOR
    WHERE TEACHER_ID = NEW_ROW.TEACHER;
    -- if prof inactive / not found, check the lecturer table
    IF is_active = FALSE THEN
        SELECT IS_ACTIVE INTO is_active
        FROM INSTANCE.LECTURER
        WHERE TEACHER_ID = NEW_ROW.TEACHER;
    END IF;
    --if inactive for both
    IF is_active = FALSE THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Cannot insert course for inactive professor or lecturer';
    END IF;
END
```

Table FACULTY

Idx	Name	Data Type	Description
An academic unit that specializes in a particular field and contains related subjects			
* Pk	ABBREVIATION	VARCHAR(5)	An identifying abbreviation for the faculty. E.g. "IT"
*	NAME	VARCHAR(100)	The full name of the faculty. E.g. "Information Technology"

Indexes

Type	Name	On	Description
Pk	PK_FACULTY	ABBREVIATION	

Table FUNCTION_PROFESSOR

Idx	Name	Data Type	Description
Responsibilities undertaken by a professor related to the management and organization of academic programs, faculty governance, and departmental operations.			
* Pk	FUNCTION_START	VARCHAR(10)	The semester in which a function starts. E.g. "WS24/25"
* Pk	PROFESSOR	VARCHAR(15)	The professor (id) which exercises this function.
* Pk	NAME	VARCHAR(100)	The name of the function. E.g. "Head of internship office"
*	LOAD	INTEGER	The hourly load of the function. E.g. "2"

Indexes

Type	Name	On	Description
Pk	PK_FUNC_PROF_NAME	FUNCTION_START, PROFESSOR, NAME	

Foreign Keys

Type	Name	On	Description
	FK_FUNCTION_PROFESSOR (PROFESSOR) ref PROFESSOR (TEACHER_ID)		

Constraints

	Name	Definition	Description
	CNS_FUNCTION	LOAD >= 0 AND LOAD <= 10	

Table LECTURER

Idx	Name	Data Type	Description
A contracted teacher, also known as "lecturer" in the project document.			
* Pk	TEACHER_ID	VARCHAR(15)	An identifying number for the teacher. Foreign Key from the teacher table.

Table LECTURER

*	ADDRESS	VARCHAR(100)	Address of the lecturer. In order to be able to write to the lecturers. E.g. "Kanalstraße 33, Esslingen"
*	IS_ACTIVE	BOOLEAN	

Indexes

Type	Name	On	Description
Pk	PK_LECTURER	TEACHER_ID	

Foreign Keys

Type	Name	On	Description
	FK_LECTURER_TEACHER (TEACHER_ID) ref TEACHER (ID)		

Triggers

	Name	Definition	Description
	TR_CHECK_EXISTING_PROFESSOR		

```
CREATE OR REPLACE TRIGGER ${nameWithSchemaName}
BEFORE INSERT ON instance.lecturer
REFERENCING NEW AS NEW_ROW
FOR EACH ROW
BEGIN
    IF EXISTS (SELECT 1 FROM instance.professor WHERE teacher_id = NEW_ROW.teacher_id) THEN
        SIGNAL SQLSTATE '45000';
    END IF;
END
```

Table MAJOR

Idx	Name	Data Type	Description
A degree program.			
* Pk	ABBREVIATION	VARCHAR(5)	An identifying abbreviation for the major. E.g. "SWB"
*	NAME	VARCHAR(100)	The full name for the major. E.g. "Software Engineering and Media Computing"
*	FACULTY	VARCHAR(5)	The faculty a major belongs to. E.g. "IT" for "Software Engineering and Media Computing"

Indexes

Type	Name	On	Description
Pk	PK_MAJOR	ABBREVIATION	

Foreign Keys

Type	Name	On	Description
	FK_MAJOR_FACULTY (FACULTY) ref FACULTY (ABBREVIATION)		

Table PROFESSOR

Idx	Name	Data Type	Description
A permanently employed academic who is responsible for teaching and research, also known as "professor" in the project document.			
* Pk	TEACHER_ID	VARCHAR(15)	An identifying number for the professor. Foreign Key from the teacher table.
*	CREDIT_HOUR_ACCOUNT	DECIMAL(5,0) DEFAULT 0	Additional work beyond the 18 compulsory work hours for a professor is credited to a credit hour account, less work is debited from this account. E.g. "2"
*	IS_ACTIVE	BOOLEAN	

Indexes

Type	Name	On	Description
Pk	PK_PROFESSOR	TEACHER_ID	

Foreign Keys

Type	Name	On	Description
	FK_PROFESSOR_TEACHER (TEACHER_ID) ref TEACHER (ID)		

Triggers

Table PROFESSOR

	Name	Definition	Description
	TR_CHECK_EXISTING_LECTURER		

```
CREATE OR REPLACE TRIGGER ${nameWithSchemaName}
BEFORE INSERT ON instance.professor
REFERENCING NEW AS NEW_ROW
FOR EACH ROW
BEGIN
    IF EXISTS (SELECT 1 FROM instance.lecturer WHERE teacher_id = NEW_ROW.teacher_id) THEN
        SIGNAL SQLSTATE '45000';
    END IF;
END
```

Table SEARCH_CRITERIA

Idx	Name	Data Type	Description
A table necessary for user views.			
*	SEMESTER_START	VARCHAR(10)	The semester for which the data is shown. E.g. "WS24/25"
*	MAJOR	VARCHAR(5)	The major for which the data is shown. E.g. "SWB"
*	SEMESTER	INTEGER	The semester of a major for which the data is shown. E.g. "6" for "SWB"

Triggers

	Name	Definition	Description
	TR_UPDATE_CREDIT_HOUR_ACCOUNT		

```
CREATE OR REPLACE TRIGGER ${nameWithSchemaName}
AFTER UPDATE OF semester_start ON INSTANCE.SEARCH_CRITERIA
REFERENCING OLD AS OLD_ROW NEW AS NEW_ROW
FOR EACH ROW
BEGIN
    -- Only if there is a difference
    IF OLD_ROW.semester_start <> NEW_ROW.semester_start THEN
        -- Call stored procedure with the new semester_start value
        CALL INSTANCE.SP_UPDATE_CREDIT_HOUR_ACCOUNT(NEW_ROW.semester_start);
    END IF;
END
```

Table SUBJECT

Idx	Name	Data Type	Description
A specific subject listed in the curriculum of a major.			
* Pk	SUBJECT_NUMBER	VARCHAR(15)	An identifying number for the subject. E.g. "R39.04995"
*	SUBJECT_NAME	VARCHAR(100)	The name of a subject. E.g. "Information Systems"
*	SUBJECT_TYPE	CHAR(1) DEFAULT 'P'	If the subject is an compulsory "P", elective "W" or supplementary "Z" E.g. "P" for "Information Systems"
*	PLANNED_STUPO_HOURS	INTEGER	The planned hours for this subject according to the StuPo. E.g. "2"
*	PLANNED_SCHEDULE_HOURS	INTEGER	The planned hours for this subject according to the schedule. E.g. "2"
	SEMESTER	INTEGER	The planned semester for this subject according to the curriculum. E.g. "6" for "Information Systems"
	MAJOR	VARCHAR(5)	The major to which this subject belongs to. E.g. "SWB"
*	FACULTY	VARCHAR(5)	The faculty of this subject. E.g. "IT"
	REMARK	VARCHAR(255)	An optional remark about the subject.

Indexes

Type	Name	On	Description
Pk	PK_SUBJECT	SUBJECT_NUMBER	

Foreign Keys

Type	Name	On	Description
	FK_SUBJECT_FACULTY (FACULTY) ref FACULTY (ABBREVIATION)		
	FK_SUBJECT_MAJOR (MAJOR) ref MAJOR (ABBREVIATION)		

Table SUBJECT

Constraints

	Name	Definition	Description
	CNS_SUBJECT	SEMESTER BETWEEN 1 AND 7	
	PSCHH_SUBJECT	PLANNED_SCHEDULE_HOURS >= 0 AND PLANNED_SCHEDULE_HOURS <= 10	
	PSTUH_SUBJECT	PLANNED_STUPO_HOURS >= 0 AND PLANNED_STUPO_HOURS <= 10	

Table TEACHER

Idx	Name	Data Type	Description
A person teaching a course.			
* Pk	ID	VARCHAR(15)	An identifying number for the teacher.
*	NAME	VARCHAR(100)	The name of the teacher. E.g. "Steffen Schober"
*	FACULTY	VARCHAR(5)	The faculty the teacher belongs to. E.g. "IT" for "Steffen Schober"
	REMARK	VARCHAR(255)	An optional remark made by the teacher about themselves.

Indexes

Type	Name	On	Description
Pk	PK_TEACHER	ID	

Foreign Keys

Type	Name	On	Description
	FK_TEACHER_FACULTY (FACULTY) ref FACULTY (ABBREVIATION)		

View VW_ACTIVE_LECTURER_WORKLOAD

```
CREATE OR REPLACE VIEW ${nameWithSchemaName} AS SELECT t.ID AS TEACHER_ID, t.NAME AS TEACHER_NAME, COALESCE(SUM(c.CREDITED_TEACHER_HOURS), 0) AS WORKLOAD FROM INSTANCE.LECTURER l JOIN INSTANCE.TEACHER t ON t.ID = l.TEACHER_ID LEFT JOIN INSTANCE.COURSE c ON c.TEACHER = t.ID JOIN INSTANCE.SEARCH_CRITERIA sc ON sc.SEMESTER_START = c.COURSE_START WHERE l.IS_ACTIVE = TRUE GROUP BY t.ID, t.NAME
```

View VW_ACTIVE_PROFESSOR_WORKLOAD

```
CREATE OR REPLACE VIEW ${nameWithSchemaName} AS SELECT t.ID AS TEACHER_ID, t.NAME AS TEACHER_NAME, p.CREDIT_HOUR_ACCOUNT, COALESCE(SUM(c.CREDITED_TEACHER_HOURS), 0) AS WORKLOAD FROM INSTANCE.PROFESSOR p JOIN INSTANCE.TEACHER t ON t.ID = p.TEACHER_ID LEFT JOIN INSTANCE.COURSE c ON c.TEACHER = t.ID JOIN INSTANCE.SEARCH_CRITERIA sc ON sc.SEMESTER_START = c.COURSE_START WHERE p.IS_ACTIVE = TRUE GROUP BY t.ID, t.NAME, p.CREDIT_HOUR_ACCOUNT
```

View VW_OFFERED_COURSES

```
CREATE OR REPLACE VIEW ${nameWithSchemaName} AS SELECT sc.SEMESTER_START, s.SUBJECT_NUMBER, s.SUBJECT_NAME, s.PLANNED_STUPO_HOURS, s.PLANNED_SCHEDULE_HOURS, s.SUBJECT_TYPE, t.FACULTY AS EXPORTING_FACULTY, s.FACULTY AS IMPORTING_FACULTY, c.ACTUAL_STUPO_HOURS, c.ACTUAL_SCHEDULE_HOURS, c.CREDITED_TEACHER_HOURS, t.NAME AS TEACHER_NAME, s.REMARK AS SUBJECT_REMARK, t.REMARK AS TEACHER_REMARK FROM INSTANCE.SUBJECT s LEFT JOIN INSTANCE.COURSE c ON c.SUBJECT = s.SUBJECT_NUMBER JOIN INSTANCE.SEARCH_CRITERIA sc ON c.COURSE_START = sc.SEMESTER_START LEFT JOIN INSTANCE.TEACHER t ON t.ID = c.TEACHER
```

View VW_OFFERED_SUBJECTS

```
CREATE OR REPLACE VIEW ${nameWithSchemaName} AS SELECT su.SUBJECT_NUMBER, su.SUBJECT_NAME, su.SUBJECT_TYPE, su.REMARK, su.PLANNED_STUPO_HOURS, su.PLANNED_SCHEDULE_HOURS FROM INSTANCE.SUBJECT su
```

View VW_PROFESSOR_FUNCTION_LOAD

```
CREATE OR REPLACE VIEW ${nameWithSchemaName} AS SELECT t.NAME AS TEACHER_NAME, f.NAME AS FUNCTION_NAME, f.FUNCTION_START, f.LOAD AS FUNCTION_LOAD FROM INSTANCE.FUNCTION_PROFESSOR f JOIN INSTANCE.PROFESSOR p ON p.TEACHER_ID = f.PROFESSOR JOIN INSTANCE.TEACHER t ON t.ID = p.TEACHER_ID JOIN INSTANCE.SEARCH_CRITERIA sc ON f.FUNCTION_START = sc.SEMESTER_START WHERE p.IS_ACTIVE = TRUE
```

View VW_SELECTED_COURSE_DETAILS

```
CREATE OR REPLACE VIEW ${nameWithSchemaName} AS SELECT sc.MAJOR, sc.SEMESTER, sc.SEMESTER_START, s.SUBJECT_NUMBER, s.SUBJECT_NAME, s.PLANNED_STUPO_HOURS, s.PLANNED_SCHEDULE_HOURS, s.SUBJECT_TYPE, t.FACULTY AS EXPORTING_FACULTY, s.FACULTY AS IMPORTING_FACULTY, c.ACTUAL_STUPO_HOURS, c.ACTUAL_SCHEDULE_HOURS, c.CREDITED_TEACHER_HOURS, t.NAME AS TEACHER_NAME, s.REMARK AS SUBJECT_REMARK, t.REMARK AS TEACHER_REMARK FROM INSTANCE.SEARCH_CRITERIA sc JOIN INSTANCE.SUBJECT s ON s.MAJOR = sc.MAJOR AND s.SEMESTER = sc.SEMESTER LEFT JOIN INSTANCE.COURSE c ON c.SUBJECT = s.SUBJECT_NUMBER AND c.COURSE_START = sc.SEMESTER_START LEFT JOIN INSTANCE.TEACHER t ON t.ID = c.TEACHER
```

View VW_SELECTED_SERVICE_DETAILS

```
CREATE OR REPLACE VIEW ${nameWithSchemaName} AS SELECT sc.SEMESTER_START, s.SUBJECT_NUMBER, s.SUBJECT_NAME, s.PLANNED_STUPO_HOURS, s.PLANNED_SCHEDULE_HOURS, s.SUBJECT_TYPE, t.FACULTY AS EXPORTING_FACULTY, s.FACULTY AS IMPORTING_FACULTY, c.ACTUAL_STUPO_HOURS, c.ACTUAL_SCHEDULE_HOURS, c.CREDITED_TEACHER_HOURS, t.NAME AS TEACHER_NAME, s.REMARK AS SUBJECT_REMARK, t.REMARK AS TEACHER_REMARK FROM INSTANCE.SUBJECT s LEFT JOIN INSTANCE.COURSE c ON c.SUBJECT = s.SUBJECT_NUMBER JOIN INSTANCE.SEARCH_CRITERIA sc ON c.COURSE_START = sc.SEMESTER_START LEFT JOIN INSTANCE.TEACHER t ON t.ID = c.TEACHER WHERE t.FACULTY != s.FACULTY
```

View VW_SELECTED_SUBJECT_DETAILS

```
CREATE OR REPLACE VIEW ${nameWithSchemaName} AS SELECT su.SUBJECT_NUMBER, su.SUBJECT_NAME, su.SUBJECT_TYPE, su.PLANNED_STUPO_HOURS, su.PLANNED_SCHEDULE_HOURS, su.SEMESTER, su.MAJOR AS SUBJECT_MAJOR, su.FACULTY AS SUBJECT_FACULTY, su.REMARK, fa.NAME AS FACULTY_NAME, ma.NAME AS MAJOR_NAME FROM INSTANCE.SUBJECT su INNER JOIN INSTANCE.FACULTY fa ON fa.ABBREVIATION = su.FACULTY INNER JOIN INSTANCE.MAJOR ma ON ma.ABBREVIATION = su.MAJOR INNER JOIN INSTANCE.SEARCH_CRITERIA sc ON sc.MAJOR = su.MAJOR  -- Join mit SEARCH_CRITERIA
```

Schema INSTANCE

Functions

F_MISSING_COURSES_CHECK

Procedures

SP_UPDATE_CREDIT_HOUR_ACCOUNT