

FS-03-066

14.04.03

GMV

Sist oppdatert: 12.06.2005

## **Notat - Integrasjon FS-Syllabus**

Det er, sammen med Scientia – utvikleren av Syllabus – utviklet et grensesnitt mellom FS og Syllabus Plus (Splus).

Kontaktpersoner Scientia:

- Matthew Finney – [matthew\\_finney@scientia.com](mailto:matthew_finney@scientia.com) +44 (0) 1223321331
- Alan Warne – [alan\\_warne@scientia.com](mailto:alan_warne@scientia.com) +44 (0) 1223321331
- [support@scientia.com](mailto:support@scientia.com)

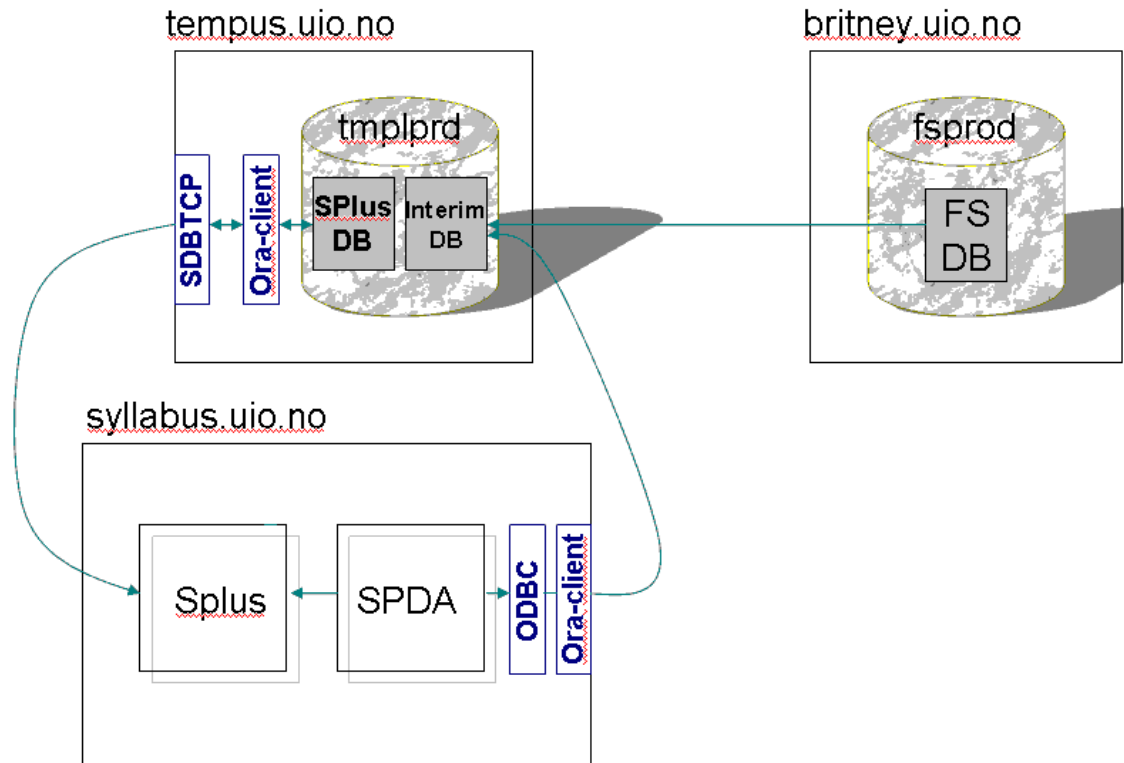
Integrasjonen mellom FS og Syllabus utføres via Syllabus Plus Data Adaptor (SPDA).

SPDA er et eget lisensiert produkt fra Syllabus. Integrasjonen består av følgende deler;

1. Det opprettes et eget skjema – InterimDB – som FS-basen og SPDA-klienten kan lese fra og oppdatere.
2. Klienten SPDA leser fra Interim-basen, og oppdaterer et Splus-image.
3. SplusDB kan da oppdateres på grunnlag av Splus-image (fra Splus-klienten).

Ved eksport fra SPlus til FS, utføres det samme, men i motsatt rekkefølge.

Figuren under viser en skisse over oppsettet ved UiO pr mai 2003.



Figur 1 Oppsett - UiO

Merk at figuren avviker fra oppsettet på et punkt, nemlig plasseringen av SDBTCP. SDBTCP (som er den programvaren som kommuniserer mellom Splus og Oracle-basen) ønskes kjørt på databasemaskinen, da det antakelig gir større stabilitet å kjøre den der. Denne er i dag installert på syllabus-serveren. Årsaken til at denne ikke er installert på base-maskinen er at base-maskinen kjører Linux. Syllabus har lovet å komme med Linux-versjon av SDBTCP, og da vil denne bli flyttet til base-maskinen.

Koblingen FS mot Interimsbasen består i en databaselink, som er koblet direkte til brukeren med interimstabellene. Fra FS-basen vil da eksport- og importrutinene kunne oppdatere og lese tabellene i interimbasen.

## Eksport fra FS

I FS-klienten er det laget et bestillingsbilde for integrasjonen mot Syllabus. Bildet inneholder funksjonalitet for hhv eksport til S+ og import fra S+. Det denne gjør, er å kjøre en rutine som oppdaterer/leser fra tabeller i interims-basen. Følgende utplukk er laget:

- Department: Henter alle steder ved egen institusjon
- Location: Henter alle bygninger, rom, romtype, romutstyrstyper og romutstyr
- Staff: Henter alle ansatte, aktiv ved en enhet (fagperson-undenhet) gitt termin
- POS: Henter alle studieprogrammer som ikke er markert utgått
- StudentSet: Henter alle emnekombinasjoner som er anbefalt, med emneinnhold.
- Module: Henter alle undervisningsenheter gitt semester, med undervisningsenheter.

FS200.050 kan kjøres som en testkjøring, der en kan angi antall forekomster en ønsker med. Denne er praktisk å benytte for å teste overføringen, da hastigheten for innlasting av data via SPDA er tidkrevende for større datamengder.

FS200.050 Integrasjon FS - Syllabus [FS200.050]

Eksport  
 Import

Termin: 2005 HØST

**Begrep**

- All
- Department
- Location++
- Staff
- POS
- StudentSet
- Module
- Activity

**Utplukk for Location**

- Alle rom
- Sted
- Rom

**Utplukk for Module**

- Alle undervisningsenheter gitt termin
- Sted
- Emne
- Generer studentsets for undervisningsaktiviteter

Slett eksisterende forekomster fra SPDA-tabeller

Testkjøring

Kun endringer i forhold til forrige kjøring (gjelder kun module/activity/templates/studentset)

Kun endring i description-felt skal ikke medføre rapportering

Med oppdatering av basen

UiO-Studieavdelingen (FS) DEMO (12.06.2005 Kl. 20:33) Side 1 av 4

**FS200.050 Integrasjon FS - Syllabus**  
Eksport til Syllabus (kun endringer) - UTEN oppdatering av DB

FS-tabell	SPlus-objekt	Gjennomgått Tatt med	
Bygning	Zone	70	70
Emne_i_Emnekombinasjon	StudentSet_Module	3737	3719
Fagperson	StaffMember	5768	5483
Fagperson-undsemester	Staff-Share	5768	5483
Planforslag	StudentSet	1540	1540

lader hentet

## Beskrivelse av utplukk fra FS for eksport til Syllabus:

### Generelt

Alle eksport-tabellene har to kolonner som er felles:

Kolonne	Konvertering	Eksempel
RecTransfer	Flag (0/1) som angir om informasjonen er lastet inn i Syllabus. Defaultverdi 0 ved overføring fra FS.	0
RecStatus	Angir om transaksjonen gjelder ny forekomst, oppdatert forekomst eller slettet forekomst. Defaultverdi U benyttes inntil videre (for oppdatering). Slette transaksjoner overføres ikke pr d.d.	U

### Department

Fyller tabellen SPDA\_Departments

Henter alle steder innenfor egen institusjon.

Kolonne	Konvertering	Eksempel
HostKey	Institusjonsnr Faknr Institutt nr Gruppenr formatert	185141200
Name	Faknr Institutt nr Gruppenr Akronym	141200 IØO
Description	Stednavn – bokmål	Institutt for østeuropeiske og orientalske studier
OwnerDepartment	Institusjonsnr Faknr Institutt nr Gruppenr formatert	185140000

### Location ++

Fyller tabellene SPDA\_Zones, SPDA\_Suitabilities, SPDA\_Locations og SPDA\_Suits\_To\_Locs.

Henter alle bygninger, og fyller tabellen SPDA\_Zones;

Kolonne	Konvertering	Eksempel
HostKey	Bygningskode	BL09
Name	Bygningskode	BL09
Description	Bygningsnavn	Eilert Sundts hus, A-blokka
Address	Gateadresse	Moltke Moes vei 31

Henter alle romtyper og romutstyrtyper, og fyller tabellen SPDA\_Suitabilities;

Kolonne	Konvertering	Eksempel
HostKey	Kode for romtype eller romutstyr	AUDS
Name	Kode for romtype eller romutstyr	AUDS
Description	Navn for romtype eller romutstyr	Auditorium

Kan velge å eksportere informasjon om alle rom, alle rom ved et gitt sted eller et bestemt rom. Informasjonen fyller tabellen SPDA\_Locations;

Kolonne	Konvertering	Eksempel
HostKey	Bygningskode Romkode	BL09132
Name	Bygningskode <blank> Romkode	BL09 132
Description	Romnavn	Auditorium 7

Department	Sted	185170000
Zone	Bygningskode	BL09
Capacity	Kapasitet – undervisning	150
Floor	Etasje	1
Comments	Merknad	

Henter alle koblinger mellom rom og romtype/romutstyr, og fyller tabellen SPDA\_Suits\_To\_Locs;

Kolonne	Konvertering	Eksempel
SuitabilityHostKey	Romtypekode / Romutstyrkode	AUDS
LocationHostKey	Bygningskode <blank> Romkode	BL09132

## Staff

Fyller tabellen SPDA\_StaffMembers

Henter alle fagpersoner som innenfor gitt termin har minst en tilknytning til fagpersonundsemester (bilde Fagperson und.semester).

Kolonne	Konvertering	Eksempel
HostKey	Fnr (11 siffer med ledende 0) for fagperson	07076930050
Name	Etternavn Fornavn	Grude Simen
Description	Stillingstittel	Amanuensis
Department	Sted – ansatt	185140700
Email	Emailadresse – person	<a href="mailto:Simen.grude@sv.uio.no">Simen.grude@sv.uio.no</a>
Address	Arbeidsadresse/telefon/fax for fagperson. Hentes fra sted dersom ikke oppgitt på fagperson	

## POS (program of study)

Fyller tabellen SPDA\_POS

Henter alle studieprogram som ikke er markert utgått.

Kolonne	Konvertering	Eksempel
HostKey	Studieprogramkode	BALTISK
Name	Studieprogramkode	BALTISK
Description	Studieprogramnavn – bokmål	Baltisk språkvitenskap magistergrad
DepartmentHostKey	Sted – fagansvarlig	185141200

## StudentSet

Fyller tabellen SPDA\_StudentSets og SPDA\_StudentSet\_Modules.

Henter alle emnekombinasjoner som er knyttet til et studieprogram som 'planforslag'. Kun studieprogram som ikke er markert utgått, tas med. Hver termin innenfor planforslaget gir et studentset.

Kolonne	Konvertering	Eksempel
---------	--------------	----------

HostKey	Emnekombinasjonskode	BALTISK_F1_1
Name	Emnekombinasjonskode	BALTISK_F1
Description	Emnekombinasjonsnavn	Forslag 1 studieplan Baltisk
DepartmentHostKey	Sted – emnekombinasjon	185141200
POSHostKey	Studieprogramkode	BALTISK

Hvert emne innenfor emnekombinasjon registreres i SPDA\_StudentSet\_Modules.

Kolonne	Konvertering	Eksempel
StudentSetHostKey	Emnekombinasjonskode	BALTISK_F1_1
ModuleHostKey	Emnekode Versjonskode	FYS114-1

## Module

Fyller tabellen SPDA\_ActivityTypes, SPDA\_Modules og SPDA\_Activities, SPDA\_Activity\_Staff, SPDA\_Activity\_Location og SPDA\_Activity\_Templates.

Henter alle undervisningsformer, og fyller tabellen SPDA\_ActivityTypes;

Kolonne	Konvertering	Eksempel
HostKey	Undformkode	LAB
Name	Undformkode	LAB
Description	Undformnavn	Laboratorieøvelse

For modules kan en velge utplukk på alle undervisningsenheter gitt semester, evt for emner begrenset til gitt sted eller for et enkeltemne. Henter undervisningsenheter som har minst en undervisningsaktivitet, og fyller SPDA\_Modules;

Kolonne	Konvertering	Eksempel
HostKey	Emnekode Versjonskode	STAGMEST-1
Name	Emnekode Versjonskode	STAGMEST-1
Description	Emnenavn – bokmål	Statsvitenskap grunnfag, metode med statistikk
Department	Sted – admin.ansvarlig	185170800
UserText	Studiepoeng, nynorsk navn, engelsk navn	Studiepoeng 12 <or> Nynorsk navn: Statsvit...
Weeks	Periode for undervisning. Hentes fra øverste undervisningsaktivitet.	23.08.2003-15.11.2003
PlannedSize	Kapasitet – hentes fra undervisningsenhet i følgende prioriterte rekkefølge: Deltakerkapasitet, Tilbud skal gis. Dersom Tilbud skal gis er større enn deltakerkapasitet, benyttes Tilbud skal gis.	270

Henter alle undervisningsaktiviteter som er knyttet til 'toppaktiviteten'. Dersom aktiviteten en ser på, har minst en timeplanforekomst som er påført klokkeslett, eller det eksisterer en aktivitet som er knyttet til denne, som har en timeplanforekomst, så vil kun denne, eller underliggende aktiviteter som er knyttet til timeplanforekomst, bli rapportert til Syllabus, da i form at Activity. Følgende blir da lagt inn i SPDA\_Activities;

Kolonne	Konvertering	Eksempel
HostKey	Emnekode Versjonskode Aktivitetskode Timeplankode	H05AST101-1-2-2-1
Name	Emnekode Versjonskode Aktivitetsnavn	AST101-1 Gruppe 2
Description	Aktivitetsnavn Disiplin	Gruppe 2 TEORI
ModuleHostKey	Emnekode Versjonskode	AST101-1
ActivityTypeHostKey	Undformkode	ØVE
DayNumber	Dagnr for ukedag	3
StartTime	Klokkeslett – starttid	14:15

Duration	Varighet - i minutter	105
Zone	Bygningskode	BL26
Weeks	Periode for undervisning. Dersom dato ikke angitt på aktivitet, arves dato fra overliggende aktivitet i aktivitetshierarkiet.	00.00.0000-00.00.0000
PlannedSize	Partikapasitet	46

Fagpersoner knyttet til aktiviteten direkte, eller via kartotek kort Fagperson, fylles inn i tabellen SPDA\_Activity\_Staff:

Kolonne	Konvertering	Eksempel
ActivityHostKey	Emnekode Versjonskode Aktivitetskode Timeplankode	H05AST101-1-2-2-1
StaffHostKey	Fnr	07066699490

Rom oppgitt i timeplan, fylles inn i tabellen SPDA\_Activity\_Location:

Kolonne	Konvertering	Eksempel
ActivityHostKey	Emnekode Versjonskode Aktivitetskode Timeplankode	H05AST101-1-2-2-1
LocationHostKey	Bygningskode Romkode	BL12310

Aktiviteter som ikke har underliggende timeplanforekomster, gir grunnlag for SPDA\_Activity\_Templates:

Kolonne	Konvertering	Eksempel
HostKey	Emnekode Versjonskode Aktivitetskode Timeplankode	H05AST101-1-2-2-1
Name	Emnekode Versjonskode Aktivitetsnavn	H05AST101-1 Gruppe 2
Description	Aktivitetsnavn Disiplin	Gruppe 2 TEORI
ModuleHostKey	Emnekode Versjonskode	AST101-1
ActivityTypeHostKey	Undformkode	ØVE
DayNumber	Dagnr for ukedag	3
StartTime	Klokkeslett – starttid	14:15
Duration	Varighet - i minutter	105
Zone	Bygningskode	BL26
Weeks	Periode for undervisning.	00.00.0000-00.00.0000
PlannedSize	Hentes fra aktivitet, i prioritert rekkefølge; Deltakere pr parti, Ønsket antall plasser, Deltakerkapasitet	46

For aktiviteter med flere timeplanforekomster kan en velge å få generert studentsets (parameteret Generer studentsets for undervisningsaktiviteter). Dette for å unngå kollisjoner mellom flere timeplanforekomster innen samme aktivitet. Tabellen SPDA\_Studentsets oppdateres:

Kolonne	Konvertering	Eksempel
HostKey	Aktivitet (UNDT_<termin><emnekode>-<versjonskode>-<terminnr>-<Aktivitetkode>	UNDT_ H05AST101-1-2-1-2
Name	Timeplan for aktivitet ...	Timeplan for aktivitet <>
Description	Timeplan for aktivitet ...	Timeplan for aktivitet <>
DepartmentHostKey	Sted – emne	185141200
POSHostKey	Gis ikke Verdi	-

Koblingen mellom studentset og aktivitet (timeplanforekomst), fylles inn i tabellen SPDA\_Studentset\_Activities:

Kolonne	Konvertering	Eksempel
---------	--------------	----------

StudentsetHostkey	Aktivitet (UNDT_<termin><emnekode>-<versjonskode>-<terminnr>-<Aktivitetkode>	UNDT_ H05AST101-1-2-1-2
ActivityHostkey	Timeplanforekomst (UNDT_<termin><emnekode>-<versjonskode>-<terminnr>-<Aktivitetkode>-<Timeplannr>	UNDT_ H05AST101-1-2-1-2-1



## Import til FS

### Generelt

Alle import-tabellene har to kolonner som er felles:

Kolonne	Konvertering	Eksempel
RecTransfer	Flag (0/1) som angir om informasjonen er lastet inn i Syllabus.	0
RecStatus	Angir om transaksjonen gjelder ny forekomst, oppdatert forekomst eller slettet forekomst. Slette-transaksjoner overføres ikke pr d.d.	U

Alle aktiviteter som importeres, kommer i tabellen FS\_Activities:

Kolonne	Konvertering	Eksempel
HostKey	Emnekode Versjonskode Aktivitetskode Timeplankode	AST101-1-2-2-1
Name	Emnekode Versjonskode Aktivitetsnavn	AST101-1 Gruppe 2
Description	Aktivitetsnavn Disiplin	Gruppe 2 TEORI
ModuleHostKey	Emnekode Versjonskode	AST101-1
ActivityTypeHostKey	Undformkode	ØVE
DayNumber	Dagnr for ukedag	3
StartTime	Klokkeslett – starttid	14:15
Duration	Varighet - i minutter	105
Zone	Bygningskode	BL26
Weeks	Periode for undervisning. Dersom dato ikke angitt på aktivitet, arves dato fra overliggende aktivitet i aktivitetshierarkiet.	00.00.0000-00.00.0000
PlannedSize	Partikapasitet	46
ActivitytemplateHostkey	Hostkey for activitytemplate denne aktivitet er generert på grunnlag av	
Period	Periodene som timeplanforekomstene gjelder for (en eller flere dato-perioder.	

Ved import av aktiviteter utføres følgende beregninger;

- Dagnr omgjøres til dagkode
- Starttime omgjøres til klokkeslett\_fra
- Klokkeslett\_til beregnes ut fra starttime og duration
- Weeks omgjøres til dato\_fra og dato\_til.

Dersom hostkey ikke starter med # og Activitytemplatehostkey ikke er gitt verdi, skal aktiviteten opprinnelig være levert fra FS. For denne gruppen aktiviteter gjøres det evt endringer i eksisterende timeplanforekomster.

Dersom hostkey starter med #, og Activitytemplatehostkey ikke er gitt verdi, så kan ikke aktiviteten lastes inn i FS, da denne timeplaninformasjonen ikke kan knyttes til noen aktivitet. Dette vil være aktiviteter som er manuelt lagt inn i S+.

Dersom HostKey starter med #, og Activitytemplatehostkey er gitt verdi, skal det lagres ny timeplanforekomst i FS, under den aktivitet som Activitytemplatehostkey viser til.

Fagpersoner knyttet til aktiviteten kommer i tabellen FS\_Activity\_Staff:

Kolonne	Konvertering	Eksempel
ActivityHostKey	Emnekode Versjonskode Aktivitetskode Timeplankode	AST101-1-2-2-1
StaffHostKey	Fnr	07066699490

Rom knyttet til aktiviteten kommer i tabellen FS\_Activity\_Location:

<b>Kolonne</b>	<b>Konvertering</b>	<b>Eksempel</b>
ActivityHostKey	Emnekode Versjonskode Aktivitetskode Timeplankode	AST101-1-2-2-1
LocationHostKey	Bygningskode Romkode	BL12310

Pr i dag hentes kun et rom fra activity-location. Omgjør locationhostkey til bygningskode og romkode (splittes).

## **SPDA-klienten**

Når eksport er utført fra FS (FS200.050), skal importen utføres i Syllabus. Det er i denne omgang ikke laget automatikk for dette, og det er ikke sikkert at det er behov for en slik automatikk her, da denne koblingen antakelig ikke skal utføres ofte, og skal utføres nokså kontrollert når det skal kjøres.

SPDA-klienten kommuniserer med interimbasen via ODBC. Oppsettet av denne er ikke tatt med i denne beskrivelsen. Før SPDA-klienten startes, må Splus-klienten startes (Splus-klienten startes evt automatisk dersom den ikke allerede er satt igang). SPDA overfører data til Splus via COM, så det er behov for at begge klientene kjører.

Splus-klienten startes med oppkobling mot en base (Scientia Database), eller kun mot et lokalt image (Standalone).

Når SPDA-klienten er startet, opprett forbindelsen med Splus ved å trykke på knappen 'Connect to Splus' nede i høyre hjørne. For å starte import fra interimstabellene, utfør menyvalg Interfave – Extract Data From Transfer Tables... Avhengig av hvilke importdefinisjoner som er registrert (se vedlegg 2), vil menyen som kommer frem inneholde et antall valg. Velg den eller de transaksjoner som ønskes utført, og trykk på 'Extract Data'.

Splus vil nå fylles med data (dersom det da ikke skjer noe uforutsett...).

Kontroller så dataene i Splus. Dersom de er OK, og skal lagres i basen, trykk på knappen WB (write back) for å oppdatere Splus-basen.

## Vedlegg 1: Tabeller i interimbasen.

### FS -> Syllabus

```
DROP TABLE SPDA_DEPARTMENTS;
CREATE TABLE SPDA_DEPARTMENTS (
  HOSTKEY varchar(255),
  NAME varchar(255),
  DESCRIPTION varchar(255),
  OWNINGDEPARTMENT varchar(255),
  RECTRANSFER number(1) default 0,
  RECSTATUS char(1) default 'U'
)
TableSpace USERS
NoLogging;
```

```
DROP TABLE SPDA_ZONES;
CREATE TABLE SPDA_ZONES (
  HOSTKEY varchar(255),
  NAME varchar(255),
  DESCRIPTION varchar(255),
  ADDRESS varchar(255),
  RECTRANSFER number(1) default 0,
  RECSTATUS char(1) default 'U'
)
TableSpace USERS
NoLogging;
```

```
DROP TABLE SPDA_LOCATIONS;
CREATE TABLE SPDA_LOCATIONS (
  HOSTKEY varchar(255),
  NAME varchar(255),
  DESCRIPTION varchar(255),
  DEPARTMENT varchar(255),
  ZONE varchar(255),
  CAPACITY varchar(255),
  FLOOR varchar(255),
  COMMENTS varchar(255),
  RECTRANSFER number(1) default 0,
  RECSTATUS char(1) default 'U'
)
TableSpace USERS
NoLogging;
```

```
DROP TABLE SPDA_SUITABILITIES;
CREATE TABLE SPDA_SUITABILITIES (
  HOSTKEY varchar(255),
  NAME varchar(255),
  DESCRIPTION varchar(255),
  RECTRANSFER number(1) default 0,
  RECSTATUS char(1) default 'U'
)
TableSpace USERS
NoLogging;
```

```
DROP TABLE SPDA_SUITS_TO_LOCS;
CREATE TABLE SPDA_SUITS_TO_LOCS (
  SUITABILITYHOSTKEY varchar(255),
  LOCATIONHOSTKEY varchar(255),
  RECTRANSFER number(1) default 0,
  RECSTATUS char(1) default 'U'
)
TableSpace USERS
NoLogging;
```

```
DROP TABLE SPDA_STAFFMEMBERS;
CREATE TABLE SPDA_STAFFMEMBERS (
  HOSTKEY varchar(255),
  NAME varchar(255),
```

```

DESCRIPTION varchar(255),
DEPARTMENT varchar(255),
EMAIL varchar(255),
ADDRESS varchar(255),
RECTRANSFER number(1) default 0,
RECSTATUS char(1) default 'U'
)
TableSpace USERS
NoLogging;

DROP TABLE SPDA_STAFF_SHARE;
CREATE TABLE SPDA_STAFF_SHARE (
  STAFFHOSTKEY varchar(255),
  DEPARTMENTHOSTKEY varchar(255),
  RECTRANSFER number(1) default 0,
  RECSTATUS char(1) default 'U'
)
TableSpace USERS
NoLogging;

DROP TABLE SPDA_MODULES;
CREATE TABLE SPDA_MODULES (
  HOSTKEY varchar(255),
  NAME varchar(255),
  DESCRIPTION varchar(255),
  DEPARTMENT varchar(255),
  USERTTEXT varchar(255),
  WEEKS varchar(255),
  PLANNEDSIZE integer,
  RECTRANSFER number(1) default 0,
  RECSTATUS char(1) default 'U'
)
TableSpace USERS
NoLogging;

DROP TABLE SPDA_ACTIVITYTYPES;
CREATE TABLE SPDA_ACTIVITYTYPES (
  HOSTKEY varchar(255),
  NAME varchar(255),
  DESCRIPTION varchar(255),
  RECTRANSFER number(1) default 0,
  RECSTATUS char(1) default 'U'
)
TableSpace USERS
NoLogging;

DROP TABLE SPDA_ACTIVITIES;
CREATE TABLE SPDA_ACTIVITIES (
  HOSTKEY varchar(255),
  NAME varchar(255),
  DESCRIPTION varchar(255),
  MODULEHOSTKEY varchar(255),
  ACTIVITYTYPEHOSTKEY varchar(255),
  DAYNUMBER integer,
  STARTTIME varchar(255),
  DURATION varchar(255),
  ZONE varchar(255),
  USERTTEXT varchar(1800),
  WEEKS varchar(255),
  PLANNEDSIZE integer,
  RECTRANSFER number(1) default 0,
  RECSTATUS char(1) default 'U'
)
TableSpace USERS
NoLogging;

DROP TABLE SPDA_ACTIVITY_TEMPLATES;
CREATE TABLE SPDA_ACTIVITY_TEMPLATES (
  HOSTKEY varchar(255),
  NAME varchar(255),
  DESCRIPTION varchar(255),

```

```

MODULEHOSTKEY varchar(255),
ACTIVITYTYPEHOSTKEY varchar(255),
DAYNUMBER integer,
STARTTIME varchar(255),
DURATION varchar(255),
ZONE varchar(255),
WEEKS varchar(255),
PLANNEDSIZE integer,
RECTRANSFER number(1) default 0,
RECSTATUS char(1) default 'U'
)
TableSpace USERS
NoLogging;

DROP TABLE SPDA_ACTIVITY_STAFF;
CREATE TABLE SPDA_ACTIVITY_STAFF (
  ACTIVITYHOSTKEY varchar(255),
  STAFFHOSTKEY varchar(255),
  RECTRANSFER number(1) default 0,
  RECSTATUS char(1) default 'U'
)
TableSpace USERS
NoLogging;

DROP TABLE SPDA_ACTIVITY_LOCATION;
CREATE TABLE SPDA_ACTIVITY_LOCATION (
  ACTIVITYHOSTKEY varchar(255),
  LOCATIONHOSTKEY varchar(255),
  RECTRANSFER number(1) default 0,
  RECSTATUS char(1) default 'U'
)
TableSpace USERS
NoLogging;

DROP TABLE SPDA_STUDENTSETS;
CREATE TABLE SPDA_STUDENTSETS(
  HOSTKEY varchar(255),
  NAME varchar(255),
  DESCRIPTION varchar(255),
  DEPARTMENTHOSTKEY varchar(255),
  POSHOSTKEY varchar(255),
  RECTRANSFER number(1) default 0,
  RECSTATUS char(1) default 'U'
)
TableSpace USERS
NoLogging;

DROP TABLE SPDA_STUDENTSET_MODULES;
CREATE TABLE SPDA_STUDENTSET_MODULES(
  STUDENTSETHOSTKEY varchar(255),
  MODULEHOSTKEY varchar(255),
  RECTRANSFER number(1) default 0,
  RECSTATUS char(1) default 'U'
)
TableSpace USERS
NoLogging;

DROP TABLE SPDA_STUDENTSET_ACTIVITIES;
CREATE TABLE SPDA_STUDENTSET_ACTIVITIES(
  STUDENTSETHOSTKEY varchar(255),
  ACTIVITYHOSTKEY varchar(255),
  RECTRANSFER number(1) default 0,
  RECSTATUS char(1) default 'U'
)
TableSpace USERS
NoLogging;

DROP TABLE SPDA_POS;
CREATE TABLE SPDA_POS(
  HOSTKEY varchar(255),
  NAME varchar(255),

```

```

DESCRIPTION varchar(255),
DEPARTMENTHOSTKEY varchar(255),
RECTRANSFER number(1) default 0,
RECSTATUS char(1) default 'U'
)
TableSpace USERS
NoLogging;

DROP TABLE SPDA_LOC_SUITS_TO_ACTIVITIES;
CREATE TABLE SPDA_LOC_SUITS_TO_ACTIVITIES (
SUITABILITYHOSTKEY varchar2(255),
ACTIVITYHOSTKEY varchar2(255),
RECTRANSFER number(1) default 0,
RECSTATUS char(1) default 'U'
)
TableSpace USERS
NoLogging;

```

## *Syllabus -> FS*

```

DROP TABLE FS_ACTIVITIES;
CREATE TABLE FS_ACTIVITIES (
HOSTKEY varchar2(255),
NAME varchar2(255),
DESCRIPTION varchar2(255),
MODULEHOSTKEY varchar2(255),
ACTIVITYTYPEHOSTKEY varchar2(255),
DAYNUMBER integer,
STARTTIME varchar2(255),
DURATION varchar2(255),
ZONE varchar2(255),
WEEKS varchar2(255),
PLANNEDSIZE integer,
ACTIVITYTEMPLATEHOSTKEY varchar2(255),
PERIOD varchar2(4000),
RECTRANSFER number(1) default 0,
RECSTATUS char(1) default 'U'
)
TableSpace USERS
NoLogging;

```

```

DROP TABLE FS_ACTIVITY_STAFF;
CREATE TABLE FS_ACTIVITY_STAFF (
ACTIVITYHOSTKEY varchar2(255),
STAFFHOSTKEY varchar2(255),
RECTRANSFER number(1) default 0,
RECSTATUS char(1) default 'U'
)
TableSpace USERS
NoLogging;

```

```

DROP TABLE FS_ACTIVITY_LOCATION;
CREATE TABLE FS_ACTIVITY_LOCATION (
ACTIVITYHOSTKEY varchar2(255),
LOCATIONHOSTKEY varchar2(255),
RECTRANSFER number(1) default 0,
RECSTATUS char(1) default 'U'
)
TableSpace USERS
NoLogging;

```

## **Vedlegg 2: Definisjoner – import SPDA.**

```

;Basic Mapping to transfer the Department information
;to S+. Hierachy is completed by using the owning department, name has the last
;six digits attached to it to ensure uniqueness as we are using the short name.
BEGIN-CLASS Department, SPDA_Departments, HostKey
HostKey,HOSTKEY
Name,NAME

```

```
Description,DESCRIPTION
Department,OWNINGDEPARTMENT
END-CLASS
```

```
;Zone information is transferred here, this links to buildings in FS
BEGIN-CLASS Zone, SPDA_Zones, HostKey
HostKey,HOSTKEY
Name,NAME
Description,DESCRIPTION
UserText1,ADDRESS
END-CLASS
```

```
;Location maps to Room data in FS, this is linked to owning department and a building,
;any information stored about floor or comments is placed in User Text fields.
;The capacity is taken as being the teaching size, we ignore the exam size of the room.
;If this size is empty a 0 value is added.
BEGIN-CLASS Location, SPDA_Locations, HostKey, ,LHF_FS_Set_Loc_Share_Post_Transfer
HostKey,HOSTKEY
Name,NAME
Description,DESCRIPTION
Zone,ZONE
Department,DEPARTMENT
Capacity,CAPACITY
UserText2,FLOOR
UserText3,COMMENTS
END-CLASS
```

```
;Suitabilities to describe the type of room.
BEGIN-CLASS Suitability, SPDA_Suitabilities, HostKey
HostKey,HOSTKEY
Name,NAME
Description,DESCRIPTION
END-CLASS
```

```
;Sets up the link between the created suitability and the room itself
;This cannot currently handle NULL columns so cannot place a suitability in this table
;that does not have a valid location
RELATIONSHIP Suitability, SUITABILITYHOSTKEY, PrimaryLocations, LOCATIONHOSTKEY,
SPDA_SUITS_TO_LOCS
```

```
;Link to the staff members
BEGIN-CLASS StaffMember, SPDA_StaffMembers, HostKey,
,LHF_FS_Set_Staff_Share_Post_Transfer
HostKey,HOSTKEY
Name,NAME
Description,DESCRIPTION
Department,DEPARTMENT
Email,EMAIL
UserText1,ADDRESS
END-CLASS
```

```
;Set up to ensure that staff are linked to all their possible departments, again like
locations
;this cannot handle NULL values
;RELATIONSHIP StaffMember, STAFFHOSTKEY, SharedWithDepartments, DEPARTMENTHOSTKEY,
SPDA_STAFF_SHARE
```

```
;Brings across basic module information, with a link to departments
;The link helper function sets the week pattern
;If the start date is blank (00.00.0000) or less than start date of institution
;then it goes from the start of term through to the end date. If the end date is blank
or
;greater than the end of term then it sets to the end of the year. If both fit in the
term
;dates then the weeks are set
BEGIN-CLASS Module, SPDA_Modules, HostKey
HostKey,HOSTKEY
Name,NAME
Description,DESCRIPTION
Department,DEPARTMENT
UserText1,USERTEXT
```



```

LHF_FS_Module_Week_Pattern,WEEKS
PlannedSize,PLANNEDSIZE
END-CLASS

;Brings in the Activity Types to define the activities
BEGIN-CLASS ActivityType, SPDA_ActivityTypes, HostKey
HostKey,HOSTKEY
Name,NAME
Description,DESCRIPTION
END-CLASS

;Activity Information. Suggested day takes a day number and sets that on in the UI
;StartTime sets the Suggested Time if it is within the Institution day
;Duration sets the number of periods for the activity, by default if this is not set
;then the duration will be 1 period
;Week pattern is the same as for modules
BEGIN-CLASS Activity, SPDA_Activities, HostKey
HostKey,HOSTKEY
Name,NAME
Description,DESCRIPTION
Module,MODULEHOSTKEY
ActivityType,ACTIVITYTYPEHOSTKEY
Zone,ZONE
PlannedSize,PLANNEDSIZE
LHF_FS_Suggested_Day,DAYNUMBER
LHF_FS_Start_Time,STARTTIME
LHF_FS_Duration,DURATION
LHF_FS_Activity_Week_Pattern,WEEKS
END-CLASS

;This is the same as Activity but placed at the Template level in S+
BEGIN-CLASS ActivityTemplate, SPDA_Activity_Templates, HostKey
HostKey,HOSTKEY
Name,NAME
Description,DESCRIPTION
Module,MODULEHOSTKEY
PlannedSize,PLANNEDSIZE
ActivityType,ACTIVITYTYPEHOSTKEY
;Zone,ZONE
LHF_FS_Suggested_Day,DAYNUMBER
LHF_FS_Start_Time,STARTTIME
LHF_FS_Duration,DURATION
LHF_FS_Activity_Week_Pattern,WEEKS
END-CLASS

;Resource requirement for staff and locations within S+ against each of the
;activities
REQUIREMENT Location, LOCATIONHOSTKEY, ACTIVITYHOSTKEY, SPDA_Activity_Location
REQUIREMENT StaffMember, STAFFHOSTKEY, ACTIVITYHOSTKEY, SPDA_Activity_Staff

;POS information for linking a Student Set to
BEGIN-CLASS ProgrammeOfStudy, SPDA_POS, HostKey
HostKey,HOSTKEY
Name,NAME
Description,DESCRIPTION
Department,DEPARTMENTHOSTKEY
END-CLASS

;Student Set is being used as the definition of a possible routh through a course
;This is linking Sets to modules
BEGIN-CLASS StudentSet, SPDA_StudentSets, HostKey
HostKey,HOSTKEY
Name,NAME
Description,DESCRIPTION
Department,DEPARTMENT
ProgrammeOfStudy, POSHOSTKEY
END-CLASS

;The link between a student set and its modules.
RELATIONSHIP StudentSet, STUDENTSETHOSTKEY, Modules, MODULEHOSTKEY,
SPDA_StudentSet_Modules

```

### Vedlegg 3: Oppsett FS - Databaselink og synonymer mot interimsbase

```
-- Følgende oppsett er nødvendig i FS-basen før installasjon av programpakken FS200.050.
-- Bruker FS - databaselink Syllabus - Interimsbase
create database link TMPLPRD Connect to SPDA_FS Identified by <passord> using
'TmplPrd.uio.no';
-- Bruker FS - synonymer - Interimsbase - alle tabeller
create synonym spda_zones for spda_zones@tmplprd;
create synonym SPDA_DEPARTMENTS for SPDA_DEPARTMENTS@tmplprd;
create synonym SPDA_LOCATIONS for SPDA_LOCATIONS@tmplprd;
create synonym SPDA_SUITABILITIES for SPDA_SUITABILITIES@tmplprd;
create synonym SPDA_SUITS_TO_LOCS for SPDA_SUITS_TO_LOCS@tmplprd;
create synonym SPDA_STAFFMEMBERS for SPDA_STAFFMEMBERS@tmplprd;
create synonym SPDA_STAFF_SHARE for SPDA_STAFF_SHARE@tmplprd;
create synonym SPDA_MODULES for SPDA_MODULES@tmplprd;
create synonym SPDA_ACTIVITYTYPES for SPDA_ACTIVITYTYPES@tmplprd;
create synonym SPDA_ACTIVITIES for SPDA_ACTIVITIES@tmplprd;
create synonym SPDA_ACTIVITY_TEMPLATES for SPDA_ACTIVITY_TEMPLATES@tmplprd;
create synonym SPDA_ACTIVITY_STAFF for SPDA_ACTIVITY_STAFF@tmplprd;
create synonym SPDA_ACTIVITY_LOCATION for SPDA_ACTIVITY_LOCATION@tmplprd;
create synonym SPDA_STUDENTSETS for SPDA_STUDENTSETS@tmplprd;
create synonym SPDA_STUDENTSET_MODULES for SPDA_STUDENTSET_MODULES@tmplprd;
create synonym SPDA_POS for SPDA_POS@tmplprd;
```