ARCADA

Databasarkivering

En kontroll av metoder för långtidsarkivering av databaser

Mathias Häggblom

Examensarbete

Medieteknik

2012

EXAMENSARBETE	
Arcada	
Utbildningsprogram:	Informations och Medieteknik
Identifikationsnummer:	3233
Författare:	Lars Mathias Häggblom
Arbetets namn:	Databasarkivering En kontroll av metoder för långtidsarkivering av databaser
Handledare (Arcada):	Johnny Biström
Uppdragsgivare:	

Sammandrag:

Databaser är det vanligaste sättet att spara samt representera stora mängder data i dagsläget. Relationerna mellan de olika tabellerna och dataposterna samt mängden olika databasformat försvårar arkiveringsprocessen. Detta examensarbete bakantar vi oss med tillgängliga tekniker för arkivering av databaser samt långtidsbevaring av dessa.

Syftet med arbetet är att betrakta de olika existerande verktygen som lämpar sig för långtidsarkivering av databaser, medan den teoretiska delen av arbetet behandlar problem som uppstår i och med långtidsbevaring, samt databasers format, historia, tekniska uppbyggnad samt arkiveringsscenarion. De arkiveringsverktyg som behandlas i detta arbete är SIARD, RODA, dataarchive och manuell insamling.

Den praktiska jämförelsen av dessa verktyg görs sedan utgående från det material som de olika organisationerna eller tillverkarna delar ut.

Nyckelord:	Databasarkivering, långtidsbevaring, databaser, arkiveringstekniker
Sidantal:	50
Språk:	Svenska
Datum för godkännande:	

DEGREE THESIS	
Arcada	
Degree Programme:	Informations och Medieteknik
Identification number:	3233
Author:	Lars Mathias Häggblom
Title:	Database archiving A review of different methods used for long-term preservation of databases.
Supervisor (Arcada):	Johnny Biström
Commissioned by:	

Abstract:

Databases are the most used way to store and represent large quantities of data nowadays. The relations and restrictions between different tables, rows and data posts make the archiving process difficult. In this thesis we familiarize ourselves with the existing models and techniques for database archiving and long-term preservation.

The purpose of this thesis is to examine the different existing tools that are suitable for long-term preservation of databases while we in the theoretical part of the thesis deal with problems due to long-term preservation and database structure, history, formats and archiving scenarios. The archiving tools or methods used in this thesis are SIARD, RODA, dataarchive and manual collection.

The practical comparison between these tools is done based on the material the different organizations or manufacturers provide.

Keywords:	Databasearchiving, long-term preservation, databases, archival technics
Number of pages:	50
Language:	Swedish
Date of acceptance:	

OPINNÄYTE	
Arcada	
Koulutusohjelma:	Informations och Medieteknik
Tunnistenumero:	3233
Tekijä:	Lars Mathias Häggblom
Työn nimi:	Tietokantoarkistointi Katsaus eri tietokantojen pitkäaikaissäilytysratkaisuihin
Työn ohjaaja (Arcada):	Johnny Biström
Toimeksiantaja:	

Tiivistelmä:

Tietokannat ovat nykyään yksi käytetymmistä tavoista suurten tietomäärien edustamisessa sekä tallentamisessa. Suhteet ja rajoitukset eri taulukoiden ja rivien välissä sekä erot eri tietokantanallejen välillä tekevät arkistoinnista vaikeaa. Tässä opinnäytetyössä tutustutaan nykyisiin malleihin ja tekniikoihin jotka käytetään tietokanta arkistointiin ja pitkäaikaissäilytykseen.

Opinnäytetyön tarkoituksena on tutkia eri välineitä jotka soveltuvat tietokantojen pitkäaikaissäilytyksen. Teoreettisessa osiossa käsitellään ongelmia jotka pitkäaikaissäilytys aiheuttaa sekä tietokannan rakennetta, historiaa, formaatteja sekä arkistointiskenaarioita. Arkistointimenetelmiä joihin perehdytään tässä työssä, ovat SIARD, RODA, dataarchive sekä tietojen käsin poimimista.

Käytännössä vertailu näiden työkalujen välillä tehdään aineiston perusteella jota eri organisaatioiden tai valmistajat tarjoavat.

Avainsanat:	Teitokantaarkistointi, pitkäaikaissäilytys, tietokanta, arkistointimenetemlmä
Sivumäärä:	50
Kieli:	Ruotsi
Hyväksymispäivämäärä:	

INNEHÅLL

Figurer

1 INLEDNING

1.1 Bakgrund

Eftersom nästan allt material som produceras i dagsläget är i någon digital form kommer det för framtiden att vara mycket viktigt att arkiveringen av detta sköts på rätt sätt. Tidigare då materialet var i pappersform kunde man enkelt arkivera det enligt gamla principer. Problemen med digitalt material är dock mycket fler och inom detta arbete skall jag försöka ta upp dem och behandla möjliga lösningar.

Nästan alla företag och organisationer har i dagsläget någon form av information i en eller flera databaser. Många av dessa databaser innehåller i arkiveringssyfte viktig information och borde därmed arkiveras. Hur skall man dock gå till väga här? Vilka regler samt standarder borde följas för att databaserna skall uppfylla kraven som ställs på arkivdugligt material?

I detta arbete bearbetas först begreppet långtidsbevaring så att läsaren skall veta vad målet är. Sedan betraktas möjliga metoder för att behandla databaser så de duger att arkiveras enligt långtidsbevaringsprinciper i digitala arkiv.

1.2 Syfte och mål

Syftet med arbetet är att jämföra metoder för långtidsbevaring av databaser samt ge en översikt av fördelar samt nackdelar för de olika metoderna.

Målet med arbetet är att kartlägga för- och nackdelarna med de olika metoderna samt hjälpa till att välja ut vilken metod som passar bäst för ändamålet.

1.3 Frågor att besvara

De frågor som jag kommer att ta upp i arbetet

- Vad bör man tänka på då en databas skall långtidsarkiveras?
- Finns det standarder för databasarkivering? I så fall hur skall dessa beaktas.
- Hur förlöper processen i praktiken då databasen arkiveras?
- Finns det färdiga lösningar? Och hur fungera dessa? Och hur skiljer de sig från varandra?

1.4 Metod

Detta arbete är en granskning av olika långtidsarkiverings tekniker.
Eftersom tester av teknikerna skulle behöva en avsevärd infrastruktur kommer arbetet inte att innehålla dessa utan basera sig på teoretiska uppgifter.
Eftersom arkiveringsverktyg som lämpar sig bäst för företagsbruk oftast inte är gratis eller öppen källkod (Open Source) så har jag tagit med dessa också i min jämförelse.

1.5 Termer och begrepp

- **SQL**

  SQL (Structured Query Language) är ett standardiserat språk för att bearbeta samt göra förfrågningar inom databaser. I dags läget är SQL det språket som de flesta databaser följer. Första SQL versionen SQL-87 kom år 1986 och sedan dess har koden utvecklats samt nya funktioner har införts. Den senaste versionen är SQL:2003 från år 2003.

- **XML**

  XML (eXtensible Markup Language) är ett märkspråk som utvecklats för att kunna dela samt flytta data mellan olika system samt program. Man kan också läsa informationen direkt ur XML filer ty datan är sparad som ren text och inte kodad.

- **Metadata**

  Metadata kan direkt översättas som ”data om data eller information”. Den vanligaste användningen är att metadatan beskriver datan exempelvis information om vad för slags data som filen innehåller, vem som skapat filen, när filen skapades eller när filen arkiverades.

- **Autenticitet**

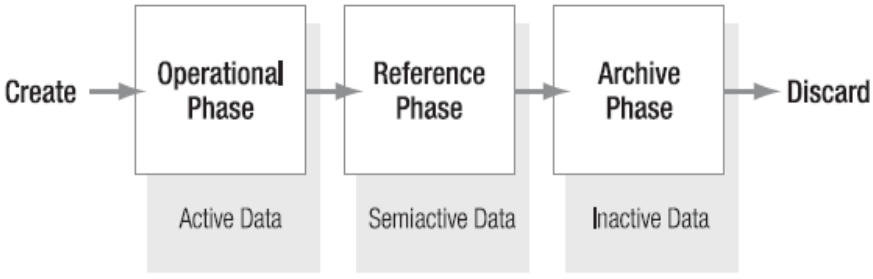
  Med autenticitet menar man att ursprunget för materialet alltid skall gå att bevisa. Man skall alltså kunna påvisa att materialet som finns i arkivet är det samma som en gång arkiverats och att materialet inte har ändrats på från originalet. I arkiven har man löst detta genom log filer som alltid visar vem som gjort något i ett arkiv och vad denna gjort.

- **Emulering och Migrering**

  För att arkivet skall vara läsligt samt användbart även i framtiden måste man antingen ändra på innehållet till nyare format så att materialet hålls läsligt (migrering) eller sedan se till att programmet som används för läsning än finns kvar och fungerar (emulering)

- **ILM**

  ILM (Information Lifecycle Management) är det läran att tillämpa vissa speciella strategier för en effektiv hantering av informationen genom hela dess livslängd. För en databas kan dessa steg ses i bilden nedan



Figur 1. ILM stegen för en databas (Olson 2009)

- **SIARD**

  SIARD (Software Independent Archiving of Relational Databases) är det schweiziska myndigheternas program för arkivering av digital data. SIARD innehåller även arkiveringen av databaser. Systemet har idag tagits i bruk även inom andra arkiverings program

- **RODA**

  RODA (Repository of Authentic Digital Objects) är det motsvarande portugisiska projektet för arkivering av digital data. Även detta program behandlar databaser.

- **DBMS**

  DBMS (DataBase Management System) eller databashanterare är programvaran som används för hanteringen av kommandon så som sökningar samt uppdateringar i databaser.

- **SIP**

  SIP (Submission Information Package) är insamligspaketen som används enligt OAIS standarden. Paketen består av tre olika delar som omfattar metadata, databasfilerna samt binärafiler med information om databasen.

- **AIP**

  AIP (Archival Information Packages) är de riktiga paketen som sparas inom ett slutförvar. I arkiveringsprocessen omvandlas SIP till AIP då inmatinginsprocessen är färdig.

- **DIP**

  DIP (Dissemination Information Package) är benämningen som används för paketen som används då man packar upp AIP och använder datan som finns i dessa.

- **METS**

  METS (Metadata Encoding and Transmission Standard) är metadatafilen i SIP. METS filen är uppbyggd enligt OAIS standarden och indelad i olika mindre delar.

- **EAD**

  EAD (Encoded Archival Description) är benämningen som används för den beskrivande delen av metadatan inom METS. EAD huvuduppgift är att hålla materialet organiserat och lättåtkomligt.

- **DBML**

  DBML (DataBase Markup Language) är en specifik typ XML fil var en relationsdatabas definieras enligt en del på förhand bestämda förutsättningar. I DBML filen försöker man bevara databasens innehåll, struktur och egenskaper.

- **BLOB**

  BLOB (Binary Large Object) är en samling binärdata i ett databas system. BLOB kan exempelvis vara bilder, ljud andra multimediaobjekt. BLOB databassupport är inte universellt.

1.6 Avgränsning

I arbetet kommer tyngdpunkten för mig att ligga på att ta reda på hur man långtidsbevarar samt arkiverar både aktiva produktionsdatabaser samt passiva databaser var hela databasen arkiveras. Ur företagssynvinkeln är arkivering av aktiva databaser mycket viktig ty den ständigt kumulativt växande datamängden. Därtill skall jag se på hur databaserna konverteras till ett dugligt format då inte vilket format som helst kan godtas samt filformaten lämpar sig för långtidsbevaring. Jag kommer inte att göra någon direkt arkiveringstest inom ramen för detta arbete. Jag kommer inte heller att koncentrera mig på hur man säkerställer sig att bevaringen hålls läsduglig under en lång tid.

Teknikerna som tas upp i detta arbeta kan användas på olika typer av databaser, jag kommer dock att koncentrera mig mest på SQL databaser eftersom dessa relationsdatabaser är de vanligaste man träffar på i företagsvälden.

2 LÅNGTIDSBEVARING

Arkivering och bevaring i sig själv är inte något nytt påhitt utan en process som funnits redan i tusentals år. Problemet i dagsläget är intåget av digitalt material som bör arkiveras. Det digitala materialet innebär att det finns så mycket mera att arkivera än då endast fysiskt material arkiverades tidigare eftersom det är lättare att producera digitalt material. Ett annat problem är de varierande datatyperna då olika typer av filer skall arkiveras på olika vis.

Med arkivering avses en metod för att bevara och beskydda material för framtida användning. Objekten man arkiverar har ofta redan levt ut sin nyttiga livslängd och de sparas endast för att tillfredsställa framtida historiska utredningar eller kuriositeter som kan förekomma. Dessutom är det viktigt att garantera tillgång till ursprunglig forskningsdata som mänsklighetens gemensamma kunskap baserar sig på.

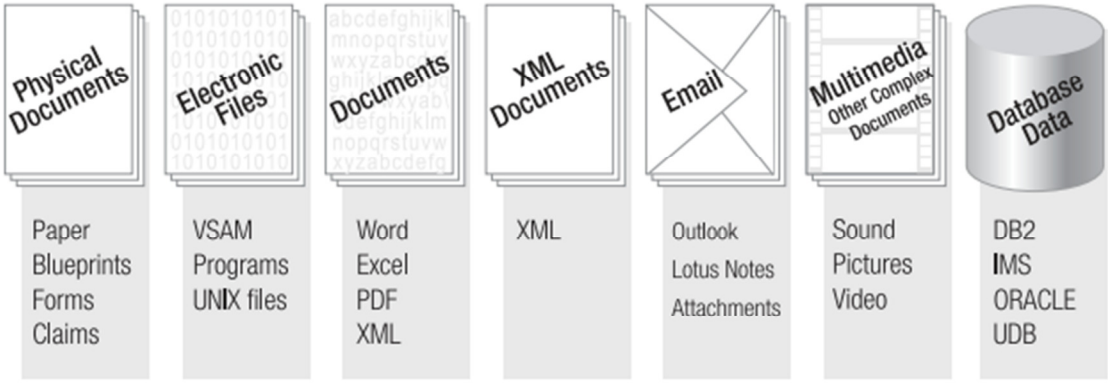
Vad som menas med uttrycket ”lång tid” i långtidsbevaring av digitala arkiv är enligt OAIS (Open Archival Information System) en så lång tid att man måste beakta nya data- samt mediaformat och att tekniken möjligen ändrat. Därtill måste man också tänka på att gamla format möjligen försvunnit samt att tekniken blivit oanvändbar. (OAIS 2008)

I långtidsbevaring skall man alltså utgå från att något som är i normalanvändning i dags läget exempelvis program eller format, kanske inte alls finns tillgängligt i framtiden. Detta igen leder till problem att läsa samt bearbeta det arkiverade materialet.

För databasers del betyder detta att DBMS system som klarar av att läsa filen inte finns tillgängliga mera. Därtill kan materialet som sparats i databasen exempelvis bilder eller video också vara av ett föråldrat format och därmed inte möjlig att ta del av.

2.1 Dataarkiveringsmodeller

I boken ”Database Archiving” från år 2009 ger författaren Jack Olson en ganska bra översikt på olika kategorier av data som arkiveras. Kategorierna som arkiveras delas upp enligt följande se Figur 2



Figur 2. Data Kategorier (Olson 2009)

2.1.1 Fysiska dokument

Den äldsta varianten av arkivering är av fysiska dokument. I dagsläget är det dock rekommenderat att konvertera de fysiska dokumenten till elektroniskt format ifall möjligt.

2.1.2 Elektroniska filer

Den enklaste varianten av digital arkivering är av elektroniska filer. Grundprincipen är att man gör en råkopia av en fil och lagrar den på en annan lagringsmedia.

2.1.3 Textdokument

Textdokumenten är ett av de viktigaste objekten inom arkivering. I arkiveringssyfte ser man på filen djupare än bara texten. Man skall spara en korresponderande metadata fil med varje textfil med identifierande information. Därtill sparas alltid information om när dokumentet skapas och modifierats i metadata filen.

2.1.4 XML dokument

Ett arkiv med XML filer brukar beaktas som själv definierande eftersom varje data element innehåller identifieringstaggar. Man måste dock komma ihåg att detta bara stämmer så länge programmet som används för att läsa XML filen förstår vad identifieringstaggarna betyder. Likt andra arkiveringsfiltyper är XML en standard som ännu utvecklas och lever så dokument skapade under en äldre variant kanske inte uppfyller alla nya standarder och därför måste uppdateras.

2.1.5 E-postkorrespondens

I dagens värld börjar e-postkorrespondens vara ett mycket viktigt arkiveringsmål. Detta ty så mycket av dagens post sköts just via e-post och vissa av dessa kan vara intressanta att spara för kommande generationer. Den egentliga arkiveringsprocessen liknar ganska långt processen för arkivering av digitala dokument. Det finns dock ännu många problem då det kommer till automatisk arkivering av e-post då skräppost och från en arkivsynvinkel onödig post filtreras bort. Ett annat problem kan vara bifogade filer av olika filtyper som måste arkiveras enligt de specifika regler som gäller för filtypen. E-postarkiv måste dessutom alltid innehålla information om ägaren av epostadressen. Ty annars kan det för an person som arbetar med arkiven i framtiden vara omöjligt att veta vem som skickade vad.

2.1.6 Multimediafiler

Med multimediafiler menas egentligen alla andra filformat. Mest handlar detta om bilder, ljud eller videofiler. Med dessa skall man vara observant med formaten som filerna sparas i och man skall följa de regler som redan finns för långtidsarkivering.

2.1.7 Databaser

Data sparad i relations-, hierarkiskt samt nätverksbaserade databaser hamnar under denna rubrik. Dessa är ofta mycket väl strukturerade filer där också relationer mellan filer bör beaktas. Databaser tillför ett unikt problem till arkiveringen i och med att man ibland vill arkivera levande databaser. Ibland vill man nämligen inte arkivera en hel da-

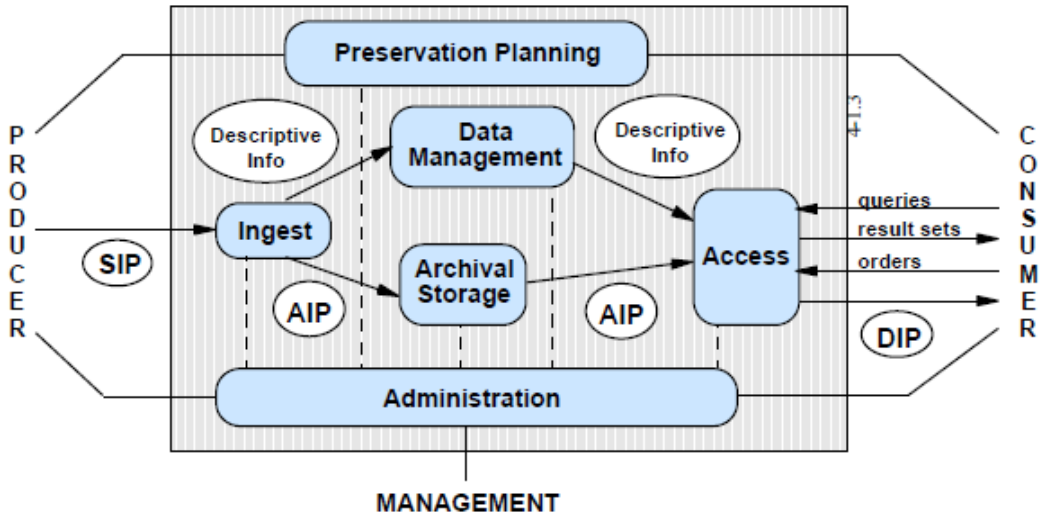
tabas utan endast plocka ut gammal data ur databasen samt arkivera denna. För metadata filerna innebär också databaser ett problem. Filen måste innehålla information om både strukturen av hela databasen samt betydelsen och relationen mellan dataobjekten. Ifall man dessutom bara arkiverar databasen delvis uppkommer ett till problem eftersom DBMS programmen ändrar med åren och därmed också metadatafilerna. Filerna är troligen liknande till format så länge man håller sig till samma databassystem. Men ifall versionerna skiljer sig för mycket eller man övergår till en annan databasstandard kan det vara problem att kombinera ihop de historiska metadatafilerna med den nyare versionen. Det sista problemet med arkiveringen av databaser kommer i vissa fall snarare att vara storleken.

2.2 Standarder

När det kommer till långtidsbevaring av digitalt material finns det ett par huvudstandarder som tillämpas.

2.2.1 OAIS – Open Archival Information System

OAIS är en samling regler eller ramverk som behandlar lagring samt hantering av digital information. I OAIS ges de begrepp och information som behövs åt ”icke-arkiv” organisationer för att dessa effektivt skall kunna medverka i arkiveringsarbetet. Ramverket ger den terminologin och informationen som behövs för planering och förståelse av framtidalångtidsarkiv. Därför används OAIS numera som en standard för digitala arkiv. (OAIS 2008)



Figur 3. Grafisk representation av OAIS funktionsprincipen (CCSD 2002)

En viktig hörnsten i OAIS ramverket är att datan som arkiveras är tudelad, en datadel och en informationsdel. I praktiken innebär detta att alla objekt består av det arkiverade materialet i digital eller fysisk form samt informationsdelen som tillåter oss att omvandla det arkiverade materialet till användbar data. För databaser betyder detta att databasen själv sparas i dataobjektet medan metadatan som vi fick då databasen arkiverades samt taggar sparas i informationsdelen. (CCSDS 2002 s. 4 – 19 f.)

2.2.2 TRAC – Trustworthy Repositories Audit & Certification (TRAC)

TRAC är en kontrollista för att bedöma tillförlitligheten och beredskapen hos instituten som gör långtidsarkivering. Arbetet är inte helt klart i dagsläget ännu men så småningom förväntar man sig ha utvecklat en fullständig revisions och certifieringskontroll lista för digitala arkiv. (CRL 2007)

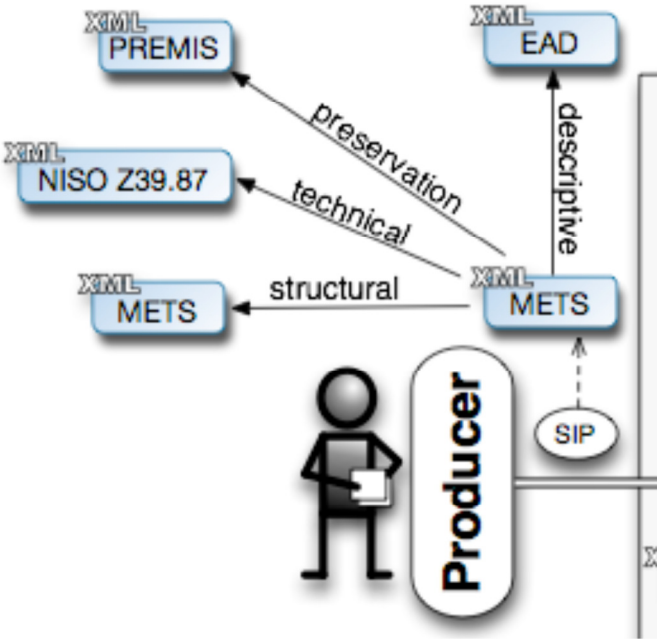
2.3 Krav

På långtidsbevarat material ställs flera krav, det handlar inte bara om att säkerställa sig om att en arkiverad fil finns kvar efter ”en lång tid”. Arkiveringsprocessen handlar lika mycket om att säkerställa sig om att filen kan läsas och bearbetas då man efter ”en lång tid” vill ta en titt på filen igen. För att dessutom kunna dra nytta av materialet måste man veta filens ursprung samt att filen inte har manipulerats under arkiveringstiden.

2.3.1 Metadata

Inom arkiveringsprocessen faller det största intresset på producent sidan av OAIS modellen. All data som arkiveras enligt OAIS modellen skall alltid också innehålla en representationsinformationsfil. Denna fil gör så att man senare utan de rätta verktygen kan göra en översättning av datan till väsentlig information.

Oberoende av vilken arkiveringstyp vi använder oss av är metadata eller beskrivande data alltid mycket viktig. Man kan på stort dela upp olika typ av metadata i mindre paket för att förstå vad denna metadata kan göra för vårt arkiv. Inom OAIS standarden samlas alla dessa metadatapaket inom en METS (Metadata Encoding and Transmission Standard) fil i ett SIP (Submission Information Package).



Figur 4. Metadatapaketets uppdelning(Ramalho 2007)

- Beskrivande metadata eller EAD (Encoded Archival Description)– för att hålla materialet organiserat och lättåtkomligt.

- Konserverings/arkiverings metadata eller PREMIS (Preservation Metadata: Implementation Strategies) – för att göra datan tillförlitlig samt för att man senare skall kunna bevisa att datansautenticitet och att den inte förändrats med tiden

- Teknisk metadata eller NISO Z39.87– är viktig för säkerheten av materialet inom arkivet.

- Strukturell metadata eller METS – är viktig för att organisera komplex digital data samt ger möjlighet att visualisera dessa objekt

Metadatan borde enligt Jack Olson beskriva strukturen samt innehållet av objekten på flera nivåer. Dessa delar är för en relationsdatabas följande:

- Dataobjekt
- Databaskolumner/element
- Databasrader
- Databastabeller
- Förhållandet/relationen mellan datan
- Datans integritetsreglerna

Relationsdatabaserna består av kolumner eller element som är grupperade i rader som sedan är samlade i tabeller. Flera tabeller kan vara beroende sinsemellan och bindas samman med exempelvis personnummer, reservdelsnummer eller postnummer. På detta vis upprepas inte data alltid i tabellerna utan man kan bygga på med information från många olika tabeller i samma sökning. Dessa relationer bör beskrivas i relationsdelen. Medan integritetsreglerna bestämmer vad fälten kan innehålla för data samt regler för denna data. Exempelvis att ditt födelsedatum måste vara ett tidigare datum än ditt anställningsdatum. (Olson 2009)

Att hitta passande material för metadatan kan vara svårt och det finns ingen universal lösning inom databasfältet. Jack Olson rekommenderar att man börjar söka i vad applikationens material ger integritetsregler samt tabellrelationerna finns oftast beskrivna här. Andra vanliga platser var man kan hitta mera metadatainformation är officiella metadatasamlingar, applikationsutvecklarens metadata samlingar, applikationskällkoden, rapporter samt människor som använder programmet.

Sedan då all denna data samlats in skall man validera denna med passliga experter inom företaget. Slutligen kan metadatan se ut som följande:

NAME	UNIT_PRICE
EXPLANATION	A numeric value that represents the intended sale price for a single unit of the product before discounts or taxes are applied. Unit prices are always assumed to be in US$.
TYPE	Decimal number with two decimal positions.
RULE 1	Cannot be zero or greater than 1000.00.

Figur 5. Exempel på vad metadatan kan innehålla för kolumn delen (Olson 2009)

TABLE NAME	ORDER_HEADER
EXPLANATION	A single row is created and stored in this table for each order received. It identifies when the order was placed, the customer, the salesperson, and current status of satisfying the order.
DATA ELEMENTS	ORDER_RECEIVED_DATE CUSTOMER_ID ORDER_NUMBER
RULE	Must be unique SALESMAN_ID STATUS
TABLE NAME	ORDER_DETAIL
EXPLANATION	A single row is created and saved for each unique item number ordered. An order can contain as many item numbers per order as desired.
DATA ELEMENTS	ORDER_NUMBER ITEM-IDENTIFIER QUANTITY_ORDERED UNIT_PRICE

Figur 6. Exempel på metadatan för tabeller (Olson 2009)

2.4 Problem

Problemen som uppstår i digitala arkiv är inte samma som man tampas med i fysiska arkiv. Därtill uppstår det också helt nya problem då man skall arkivera en fil för en lång tid i jämförelse med korttidslagring. (Digital Preservation Europe. What is digital Preservation? 2006)

2.4.1 Äkthet

I framtiden då materialet som arkiverats skall användas måste personen veta vad materialet är samt vad det handlar om. Därtill måste man kunna säkerställa sig om att materialet inte har manipulerats under den tiden det varit arkiverat utan att materialet är det som en gång arkiverades. Om materialet har manipulerats avsiktligt exempelvis vid migrering till ett nyare format måste detta vara noga bokfört så att man kan ta reda på när det skett samt vem som manipulerat materialet. Enligt OAIS standarden måste varje arkiverad fil ha en sådan här referensfil. Dessa är liknade problem som finns i traditionella fysiska arkiv med eftersom man också här måste bokföra ifall materialet manipuleras samt att källorna skall förekomma. (Factor 2009)

2.4.2 Användbarhet

I användbarheten skiljer sig digitala arkiv från de fysiska arkiven mera och här uppstår specifika problem som endast berör det digitala arkiven. Man kan lättast förklara detta med ett litet jämförande exempel. Vi räknar här med att både det fysiska– och digitala arkiven klarar sig undan olyckor som naturkatastrofer eller liknade. Vi börjar med att enkelt stoppa in en katalog som får motsvara en databas i ett fysiskt arkiv och en databasfil i ett digitalt arkiv. Sedan väntar vi exempelvis 150 år och plockar ut bägge, katalogen som sparats i det fysiska arkivet kan nu ännu läsas utan problem. För den databasfilen är målet inte lika lätt att nå utan flera hinder finns i vägen. (Digital Preservation Europe 2006)

- **Lagringsmediet**

  Dagens digitala lagringsmedier kan man inte beskriva som väldigt hållbara i jämförelse med pappret i katalogen. Hårdskivor kan gå sönder eller data bli oläsligt av en liten törn eller då de kommer i kontakt med magnetfält likaså kan band ur bandstationer bli sköra med tiden och gå sönder av fysisk påfrestning. CD eller DVD skivor är heller inte media man kn lita på eftersom enkelt repas och kan förstöras av bara ljus därtill vet vi inte med säkerhet hur de reagerar då de blir väldigt gamla. Visst blir pappret också skört med tiden men så länge temperaturen och luft fuktigheten hålls på passlig nivå förstörs inte papper under denna tid. (Digital Preservation Europe 2011)

- **Hårdvaran**

  Det finns inget som säjer att just den typen av hårdvara som läser lagringsmedian vi sparade vår databasfil på finns tillgänglig i framtiden. Speciellt ifall vi använder oss av udda format kan det vara att det inte finns något som kan läsa filen, detta fastän lagringsmedian inte skulle ha förstörts under arkiveringstiden. Det fysiska dokumentet har inga sådana här problem över huvud taget. (Digital Preservation Europe 2011)

- **Filformatet**

  Ifall nu lagringsmedian hållits hel under arkiveringsperioden och vi lyckats hitta hårdvara som kan läsa median måste vi ännu kunna öppna själva filen. Här kommer problemet med att filformat föråldras snabbt in, inget säjer att framtidens program klarar av att vara bakåtkompatibla med så pass gamla filer och därmed är den arkiverade filen oanvändbar. Sådana här problem uppstår redan i dagsläget med filer som endast är högst 30 år gamla. Då kan man tänka sig att problemet blir mycket värre då man lägger till över 100 år av utveckling.

De digitala arkiven kräver alltså en hel del mera tillsyn än de fysiska arkiven. Man måste regelbundet hålla koll på att lagringsmedian ännu är hel, att det finns hårdvara som kan läsa den och att det ännu finns program som kan bearbeta filen. Redan i dagsläget kan filer som är 10 – 15 år gamla skapa problem för moderna program. (Digital Preservation Europe 2011)

Då tekniken går framåt måste man i praktiken två alternativ. Man kan antingen följa upp den nya tekniken och flyttar materialet till exempelvis ny hårdvara, lagringsmedia samt spara filen i ett nytt format. Denna praxis kallas allmänt för migrering. Det andra alternativet kallas sedan för emulering. Detta innebär att man håller kvar den gamla tekniken a genom att köra dem i exempelvis virtuella maskiner. Programvara och hårdvara kan möjligen bevaras genom emulering men för lagringsmedian i sig själ är migrering det bästa alternativet för att försäkra sig om att median inte går sönder. I dagsläget är migrering den praxis som allmänt rekommenderas. (Long 2009)

3 DATABASER

3.1 Definition och historia

Runt år 1964 myntades termen ”Databas” för att beteckna samlingar av data som delas av slutanvändare ett tidsdelningsdatorsystem. Med termen menades iden med att flera program kunde spara och dela på information i endast en fil. Tidigare hade varje program sin egen ”master file” med data nämligen. Detta medförde en klar fördel i effektivitet. (DuCharme 2005)

Tidiga versioner av databaser var IBM’s IMS (Informations Management System) utgivet år 1968 samt Cullinet Softwares IDMS presenterat under tidigt 70-tal. IMS byggde på en hierarkisk modell där informationen fanns lagrad i en trädstruktur. Detta gjorde informationen snabb att hitta men svår att manipulera. IDMS igen byggde på en nätverksdatamodell, detta betyder att den redan liknar mera dagens relationsdatabasmodeller. Dessa två typer har dock med åren blivit föråldrade och används inte mera. (O'Neil 2001)

3.1.1 Relationsdatabaser

Relationsdatabasmodellen introducerades av Edgar Codd ca 1970. Skillnaden till de tidigare databasmodellerna var att man här sparar all data i tabeller där datan kan höra samman på flera olika vis enligt n: m principen. Relationsdatabasmodellen införde också användningen av primär- och sekundärnycklar samt unika nycklar. Dessa hjälper till att minska på upprepningen av samma data på flera ställen i databasen. (Bernstein 2008) (Descartes 2000)

Eftersom över 90 % av alla databaser i dagsläget är av relationsdatabasmodell betyder det att man i arkiveringsarbetet kan koncentrera sig på dessa.

3.1.2 Objektdatabaser

Under 1980 och 1990 talets början trodde man att objektorienterade databaser skulle vara framtidens melodi. I en objektorienterad databas är datan representerad i objektsform typ liknade som används inom objektorienterad programmering. Dessa slog dock aldrig riktigt igenom ty de är långsammare än relationsdatabaser i typiska databassökningar. Under de senaste åren har de objektorienterade databaseran upplevt något av en andra uppswing och kan möjligen i framtiden vara något man måste beakta då bevaringsplaner utvecklas. (Hand. 1998) (Wikipedia Object database, 2012)

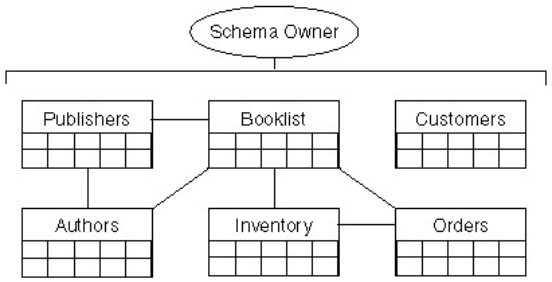
3.2 Databasdelarna

För att förstå vad som behövs beaktas vid långtidsarkiveringsprocessen av en databas måste vi ta upp vissa viktiga delar som dessa består av. Här nedan ser vi en lista på sådana plockade ur Ryan Stephens bok Database Design från år 2000

- **Dataelement**
  Byggstenarna i databas är kolumner eller element. Varje sådant här element borde ha ett namn samt en förklaring över vad värdet beskriver inom databashelheten

- **Databastabeller**

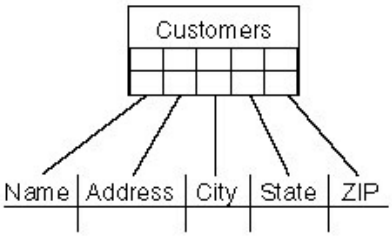
  Tabeller är primärskiktet av en databas var all information sparas. Databasrader men samma tabelldefinition grupperas i tabeller. Här kommer relationsaspekten för relationsdatabaser. När en användare söker något i en databas är det via dessa som man orienterar sig.



Figur 7. Databastabeller visualiserat (Stephens 2000)

- **Databaskolumner**

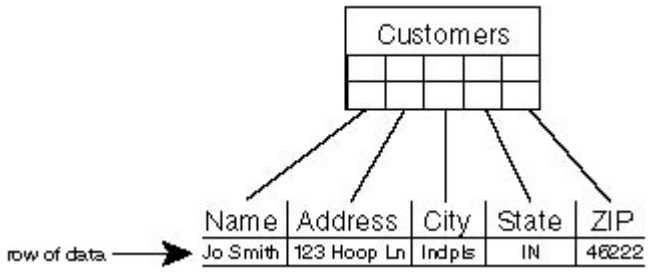
  Databaskolumner är nästa skikt i databasen. Man kaShemann säja att en kolumn representerar en del av databasen var all information hör ihop.



Figur 8. Databastabeller visualiserat (Stephens 2000)

- **Databasrader**

  Flera databaselement grupperas i rader för att beskriva en lite större del samt sammanhanget mellan olika dataelement. Förenklat betyder det att en rad data motsvara en datapost i systemet.



Figur 9. Databasrader visualiserat (Stephens 2000)

- **Datatyper**

  Datatypen bestämmer vad för data man kan spara i en kolumn i databasen. Det finns flera olika typer och olika begränsningar för dessa. Exempel på de vanligaste är kanske ”Alphanumeric”, ”Numeric” och ”Date and time”

- **Nycklar**

  Primärdatan är märkt med olika typer av nycklar exempelvis primärnycklar så kallade Primary Keys samt sekundärnycklar så kallade Foreign Keys. Dessa nycklar garantera integriteten för databasen dvs. vilka tabeller som hänger ihop samt vilka fält är unika i en databas.

- **Aktiva regler eller Triggers**

  Triggers är automatiserade händelser eller operationer som databashanteraren utför automatiskt då något på förhand bestämt villkor uppfylls eller operation körs i en databas.

- **Vyer**

  Vyer är på förhand sparade SQL frågor dvs. queryn. Resultatet är egentligen en virtuell tabell där innehållet räknas ut på nytt varje gång vyn uppdateras.

- **Lagrade procedurer (engelska: stored procedure)**

  En lagrad procedur är färdiga rutiner som finns sparade direkt i databasen. I vissa fall är dessa procedurer skrivna i Java, C eller direkt i SQL kod. Dessa procedurer kan sedan köras automatiskt så fort vissa villkor uppfylls.

- **Användare samt roller**

  En databas har oftast en mängd olika användare med olika rättigheter. Olika användare kan därtill vara uppdelade i olika roller. Rollerna kan definieras så att vissa roller enbart kan läsa innehåll och andra roller redigera eller lägga till data i databasen. Detta betyder också att alla användare inte direkt kan se allt i en databas.

- **Scheman**

  Scheman är en samling av tabeller, vyer och rutiner. Ett schema ligger innanför en databas

4 ARKIVERINGSSCENARION

I databasarkivering kan det finnas ett par olika scenarion som måste beaktas. I det första scenariot vill man arkivera en hel databas så som den är. Scenario två är det vanligare åtminstone inom företagsvärlden. Här arkiverar man en aktiv produktionsdatabas eller en del av denna. Långtidsbevaring är viktigt att kunna uppnå i bådadera scenariona. Vägen som man måste följa kan dock aningen skilja sig mellan dessa två. (Grid-tools 2010)

4.1 Arkivering av en hel databas

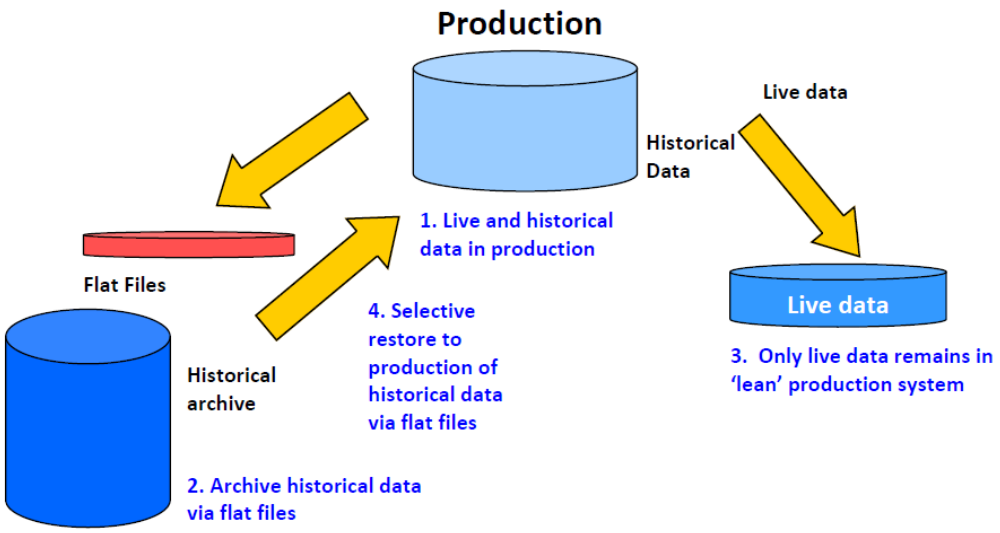
Då man talar om direkt långtidsarkivering är det här det lättare och mera stödda scenariot. Här tar man alltså en hel inaktiv databas eller en ögonblicksbild av en databas och arkiverar denna. Det är denna typ av arkiverings modell som stöds av både RODA och SIARD. (Factsheet SIARD 2010) (SIARD Format Description 2009)

4.2 Arkivering av en aktiv databas

På grund av att gammal data ofta hänger kvar i företagsdatabaser håller vissa på att växa till allt för stora i dagsläget. Därför kommer denna typ av arkivering att bli mycket essentiell inom en snar framtid. Man kan dock inte gå till väga hur som helst med arkiveringen här ty för företagen gäller det att följa en del olika lagar och förordningar.
(Olson 2009 s. 21)

En möjlighet är att skapa en rutin som plockar ut gammal data enligt på förhand bestämda villkor och flyttar denna data till en egen arkiveringsdatabas som sedan kan behandlas på samma sätt som i scenario ett där hela databasen sedan arkiveras.
(Olson 2009 s. 11)

En annan möjlighet är att satsa på något av de specifika kommersiella system som gör denna arkivering automatiskt. Ett av dessa verktyg är ”Dataarchive by Grid-tools”. Verktyget gör så att ett företag mer eller mindre automatiskt kan dela upp en aktivdatabas i olika typer av material som kan arkiveras eller hållas aktivt.
(Grid-tools 2010)



Figur 10. Strukturen för dataarchive(Grid-tools 2010)

Genom att införa en del arkivering av en Aktiv produktionsdatabas kan företaget se fördelar i att prestandan blir bättre. Detta ty den totala mängden data en sökning måste gå igenom minskar. Därtill sparar också företaget utrymme i och med arkiveringen då arkiverad data bara tar en bråkdel av samma utrymme som aktiv data.
(Grid-tools 2010)

5 METODER FÖR DATABASARKIVERING

Då man arkiverar databaser görs detta ofta med en del färdigt utvecklade verktyg. Härnäst har jag tagit upp några av databasarkiveringsverktygen.

Inom databasarkiveringen sker arkiveringsprocessen oftast utgående från databasmaterialet och inte från DBMS verktyget. Istället har de olika arkiveringsverktygen egna specifikt utvecklade verktyg som sköter om inläsningen av data. Det finns dock också möjligheten att arkivera en databas som inte det direkt finns stod för inom RODA eller SIARD genom att manuellt arkivera filen i flat file format. Detta är dock en manuell process som kräver stort tekniskt kunnande.

5.1 Flat file format

Flat file format beskriver dem mest rudimentära typen av databasarkivering. Här sparas varje tabell som en egen textfil enligt gällande arkiveringsstandarder för vanliga textfiler. I textfilen sparas alla rader ur en tabell på egna rader i ett textdokument med på förhand bestämda avgränsare typ tab eller kommatecken. Inom flat file databaser finns det inga relationer mellan filerna direkt utan allting måste beskrivas i en tillhörande metadata fil. Det finns inte heller några beskrivningar på vad för krav som gäller för de olika tabellerna exempelvis datatyper eller fältlängder färdigt specificerade. Detta måste man själv spara i den beskrivande metadatafilen ty annars kan man inte senare bygga upp databasen på nytt så att den ser likadan ut som då den arkiverades.
(Olson 2009)

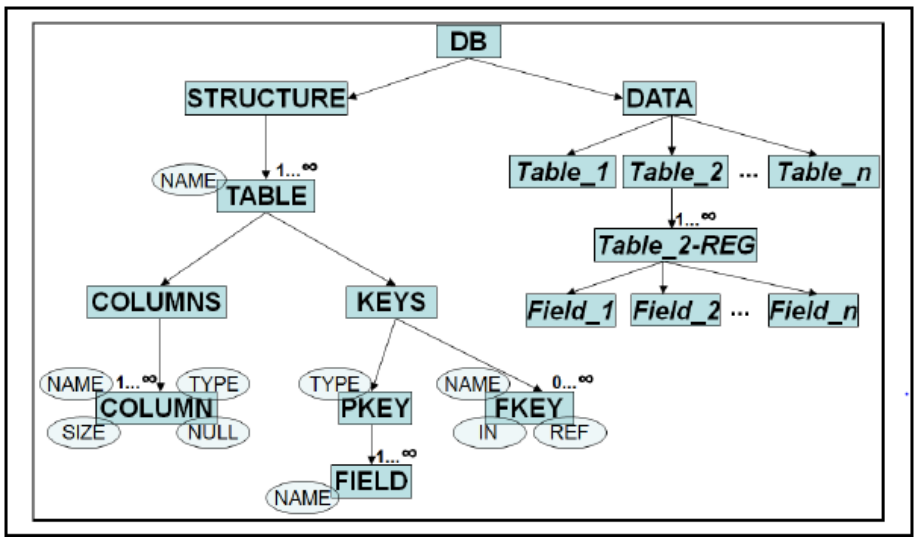
Fördelarna med flat file format databaser är att det inte egentligen finns restriktioner för hurdana databaser man kan arbeta med till skillnad från exempelvis SIARD samt RODA.

Grunden till de mera utvecklade databasarkiveringsverktygen ligger dock i flat file databasarkivering och funktionsprincipen är både i SIARD och RODA i praktiken den samma. Skillnaden ligger bara att dessa två standarder utvecklat färdiga verktyg för omvandlingen av datan till data filer samt tillhörande metadatafiler. Här sker dock allt arbete manuellt och den som arkiverar måste ha en mycket god kunskap.
(Olson 2009 s. 64)

5.2 RODA

RODA (Repository of Authentic Digital Objects) är det Portugisisk projektet för att preservera digitalt material. Projektet följer OAIS reglerna för långtidsarkivering och behandlar också databasfiler. RODA använder migreringsprincipen som huvudsaklig arkiveringsstrategi. Med detta menas att man inom projektet omvandlar de olika databasfilerna till DBML filer. DBML filerna är en specifiktyp av XML filer utvecklade enligt ett eget schema av projektdeltagarna som beskriver både datan och strukturen av databasen. I bilden under ser vi först hur strukturen för DBML filen är uppbyggd i RODA.
(Faria 2009) (Ramalho 2011)

5.2.1 Arkivering med RODA



Figur 11. DBML filens struktur (Freitas2011)

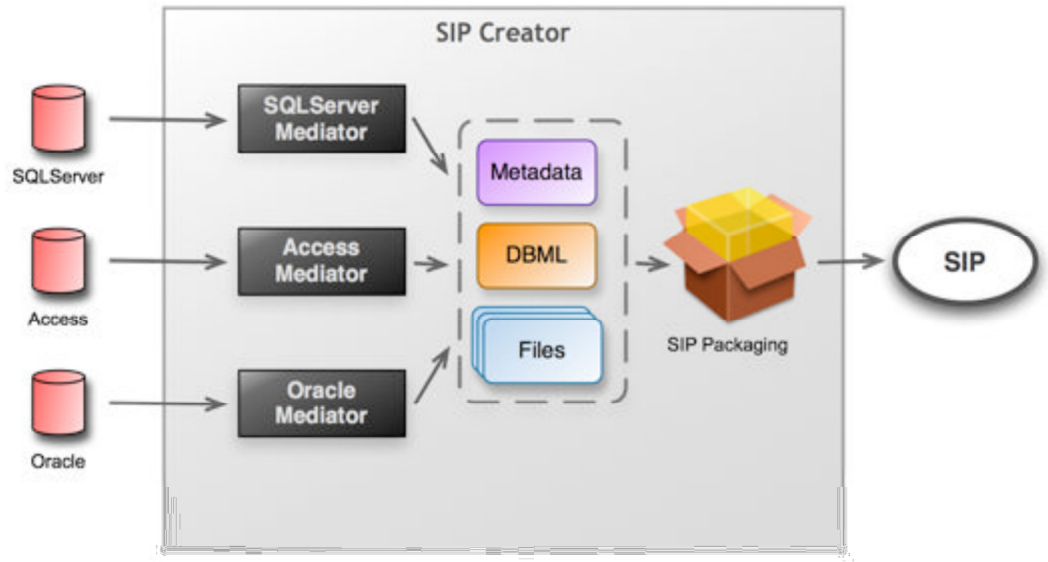
För att förstå hur DBML filen på riktigt ser ut inom RODA finns ett exempel här under för en exempel tabell.

```
<?xml version="1.0" ?>
 <DB>
  <STRUCTURE>
   <TABLE NAME="products">
    <COLUMNS>
     <COLUMN NAME="code" TYPE="nvarchar"
      SIZE="10" NULL="no"/>
     <COLUMN NAME="description" TYPE="nvarchar"
      SIZE="50" NULL="no"/>
     ...
    </COLUMNS>
    <KEYS>
     <PKEY TYPE="simple">
      <FIELD NAME="code"/>
     </PKEY>
    </KEYS>
   </TABLE>
   <TABLE NAME="p2s">
    <COLUMNS>
     <COLUMN NAME="cod-p" TYPE="nvarchar"
      SIZE="10" NULL="no"/>
     <COLUMN NAME="cod-s" TYPE="nvarchar"
      SIZE="10" NULL="no"/>
    </COLUMNS>
    <KEYS>
     <PKEY TYPE="composite">
      <FIELD NAME="cod-p"/>
      <FIELD NAME="cod-s"/>
     </PKEY>
     <FKEY NAME="cod-p" IN="products"
      REF="code"/>
     <FKEY NAME="cod-s" IN="suppliers"
      REF="code"/>
    </KEYS>
   </TABLE>
   <TABLE NAME="suppliers">
    <COLUMNS>
     <COLUMN NAME="code" TYPE="nvarchar"
      SIZE="10" NULL="no"/>
     <COLUMN NAME="name" TYPE="nvarchar"
      SIZE="60" NULL="no"/>
     ...
    </COLUMNS>
    <KEYS>
      <PKEY TYPE="simple">
        <FIELD NAME="code"/>
      </PKEY>
    </KEYS>
   </TABLE>
  </STRUCTURE>
  <DATA>
    ...
  </DATA>
</DB>
```

Figur 12. DBML filens uppbyggnad (Ramalho 2007)

Man kan enkelt se att alla delar av tabellen är representerade samt att det finns en beskrivning för hur tabellen skall läsas samt vilka krav det finns för datan i fälten.

För tillfället Stöder RODA MSSQL Server, MySQL, Oracle samt Microsoft ACCESS databaser men utbudet utvecklas enligt behovet. För att långtidsarkiveringskraven skall uppnås körs de olika typerna av databaser genom en konverterare som RODA gruppen skapat. Konverteraren skapar sedan SIP filen enligt de standarder som OAIS definierar.



Figur 13. SIP packet i RODA (Ramalho 2007)

SIP filerna består av följande delar. För det första består den av en METS metadata fil. Denna METS fil innehåller enligt OAIS principerna en EAD metadata fil och som beskriver databasen, den slutliga strukturen för denna EAD fil inom RODA projektet är ännu under utveckling. Därtill innehåller paketet DBML filerna som representerar den ursprungliga databasens struktur samt innehåll. Som sista del i SIP paketet finns en knippe binära filer som motsvarar BLOB filer (binary large objects) som vi kan hitta inom den ursprungliga databasen. (Ramalho 2007)

AIP (Archival Information Packages) är de riktiga paketen som sparas inom slutförvaret. Dessa AIP paket kan inte direkt sedan behandlas utan måste packas upp tillbaka för att en användare skall kunna ta del av det. (Ramalho 2007)

I RODA projektet har experterna kommit fram till att då det gäller att hålla arkiverade databaser läsliga under en lång tid måste man utveckla ett slutförvar som är kapabel att spara abstrakta representationer av databasen. Dessa abstrakta representationer möjliggör för att man kan separera datan från strukturen och på så sätt få en databas som inte är beroende av någon viss DBSM.

Härmed går man miste om alla funktionaliteten som den ursprungliga databasen hade exempelvis sökning samt möjligheten att köra vyer eller likande funktioner. Det man dock vinner är att både datan och strukturen samtidigt hålls läsbar samt oförändrad genom att allt sparas i XML filer.

Databaserna som arkiveras inom RODA projektet är alltid frysta databaser dvs. databaser som ej mera används eller ögonblicksbilder av databaser från en viss tidpunkt. På detta vis stöds varken uppdatering eller nya inlägg i databaser.

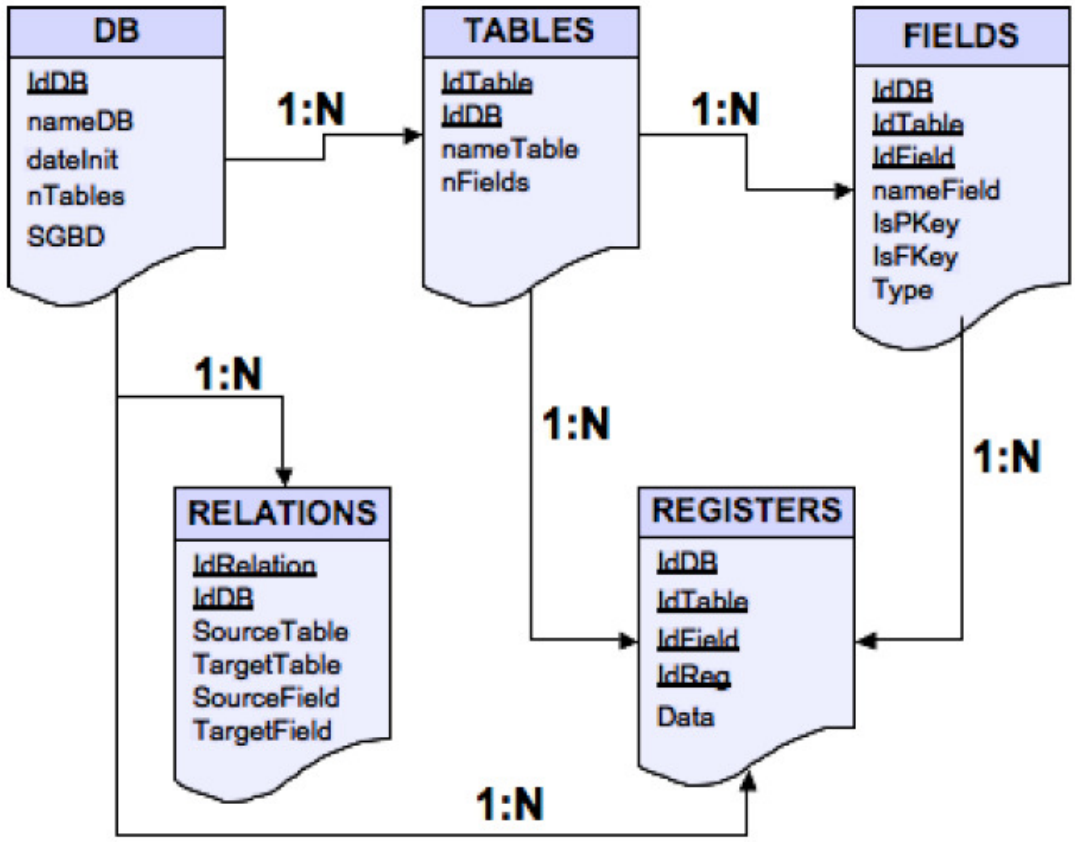
5.2.2 Användning av den arkiverade datan

Den vanliga användare som skall arbeta med arkivet kommer inte vara intresserade av en enda stor XML fil som representerar vad de förväntar sig att vara en relationsdatabas. För att lösa denna fråga har experterna inom RODA projektet utvecklat två olika utmatningsprocesser: SQL och HTML.

Figur 14 Databas exportering (Ramalho 2007)

Antingen kan användaren exportera en stor SQL fil ur AIP paketet som sedan kan importeras i en DBMS och sedan använda denna för att komma åt datan och göra sökningar inom den.

HTML exporten skapar igen en dynamisk databas bläddrare som tillåter användarna att bläddra i databasen. Ifall databaserna är mycket stora sätter de vissa begränsningar på denna HTML bläddring. Därför har RODA teamet infört ett Cache system i en tillfällig databas. Eftersom det är meningen att cache databasen skall kunna stöda olika typer av databassystem vi var teamet tvungna att skapa en egen abstrakt relationsmodell som passar för detta.



Figur 15 Databas cache modellen(Ramalho 2007)

Som man ser ur figuren ovan finns egna tabeller för att lagra databasinformationen, tabellinformation, fältinformation och data ("register"). Alla dataposter omvandlas till textform för att de skall kunna presenteras i denna tabell.

(Ramalho 2007)

5.3 SIARD

SIARD eller “Software Independent Archiving of Relational Databases” är ett långtidsbevaringsformat utvecklat av de Schweiziska nationella arkiven. I dagsläget har också SIARD blivit vald som det officiella långtidsbevaringsformatet för databaser inom det sameuropeiska PLANETS ”Preservation and Long-term Access through Networked Services” projektet. Formatet i sig själv bygger på att spara filerna i endast öppna standarder. Det innehåller en blandning av unicode textfiler, XML, SQL1999 och Zip filer.
(Factsheet SIARD 2010)

Den största svagheten för tillfället är att SIARD Suite endast kan arkivera databaser som är av formatet SQL, Oracle eller Microsoft Access. Dessa tre är dock de mest använda typerna av databaser så orsaken varför just dessa tre valts är motiverad. På grund av filformatet som de arkiverade databaserna sparas i lämpar sig alltså SIARD väl för långtidsarkivering.
(SIARD Format Description 2009)

5.3.1 SIARD Suite verktygen

”SIARD Suite” är den egentliga verktygsbacken som används för att läsa in och bearbeta databaserna. Verktygen är alla kodade i Java vilket betyder att de fungerar i alla operativsystem som finns på marknaden i dagsläget. Följande program hör hit:
(Factsheet SIARD 2010)

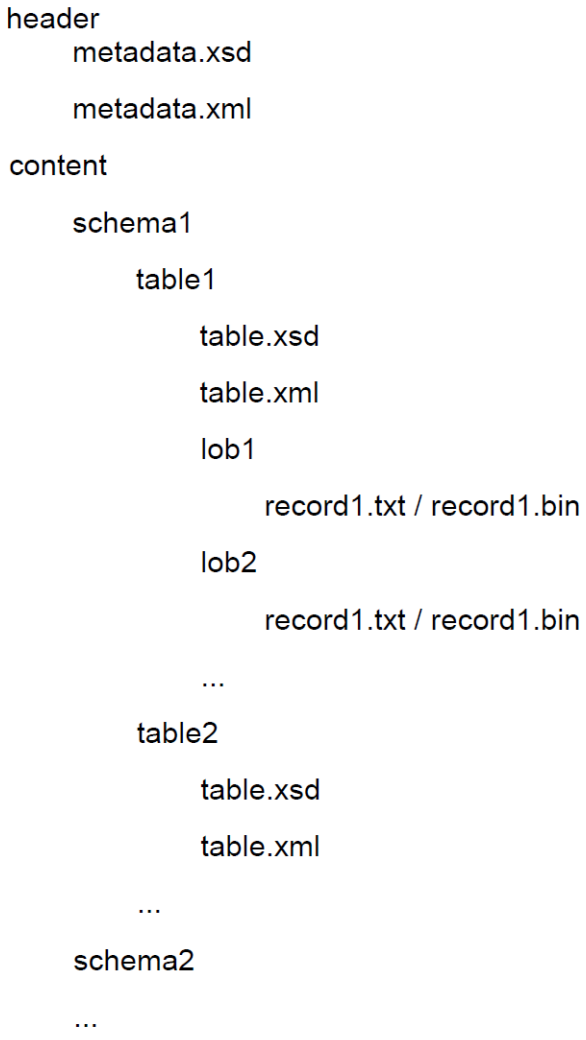
”SiardFromDb” är migreringsverktyget som används för att läsa in databasen samt konvertera den till SIARD formatet.

”SiardEdit” är ett verktyg som tillåter användaren att editera samt uppdatera metadata filerna. Programmet tillåter också användaren att göra sökningar inom metadatan och därmed inom den arkiverade databasen. Ifall man endast behöver kontrollera något ur databasen kan detta verktyg vara tillräckligt.

”SiardToDb” är den tredje delen i paketet. som namnet säjer tillåter detta packet användaren att ladda in SIARD filer i någon av de tre stödda databassystemen. På samma sätt ger detta alltså en möjlighet att migrera data från en typs databas till en ny. Detta igen ger möjligheten att fritt editera och göra sökningar i databasen igen.

5.3.2 Strukturen för SIARD data

Arkivstrukturen för SIARD data är delad i flera delar. Den första delen består av en gemensam XML metadata del som beskriver strukturen på de arkiverade databaserna, därtill ger den information om var primärdatan hittas i arkivet. De andra delarna är de primära delarna som innehåller databasmaterialet. Varje databas har alltså sin egen XML fil. Tillsamman sparas sedan dessa två delar i en gemensam ”.SIARD” fil. Filen är ett okomprimerat ZIP-arkiv där metadata delen finns i file header delen samt primärdatan i contet delen enligt följande princip.
(SIARD Format Description 2009)



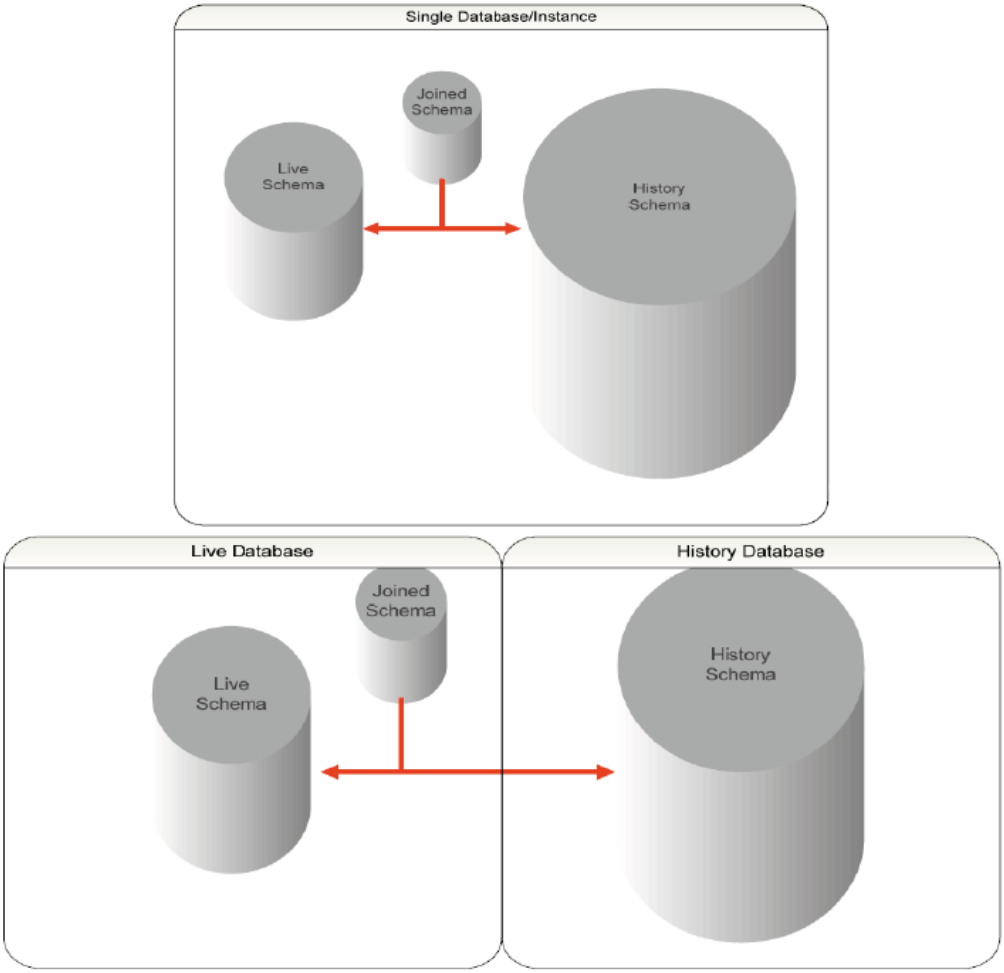
Figur 16. XML filen i SIARD (SIARD Format Description 2009)

5.4 Dataarchive av Grid-tools

Dataarchive är en kommersiell produkt som lämpar sig för exempelvis företagsbruk. Produkten kan behandla databaser exempelvis Oracle, DB2, MySQL, SQL Sybase, Filestore samt ViewDirect. Själva arkiveringsprocessen är aningen olika mellan de olika typerna men för en SQL/MySQL databs är funktionen följande

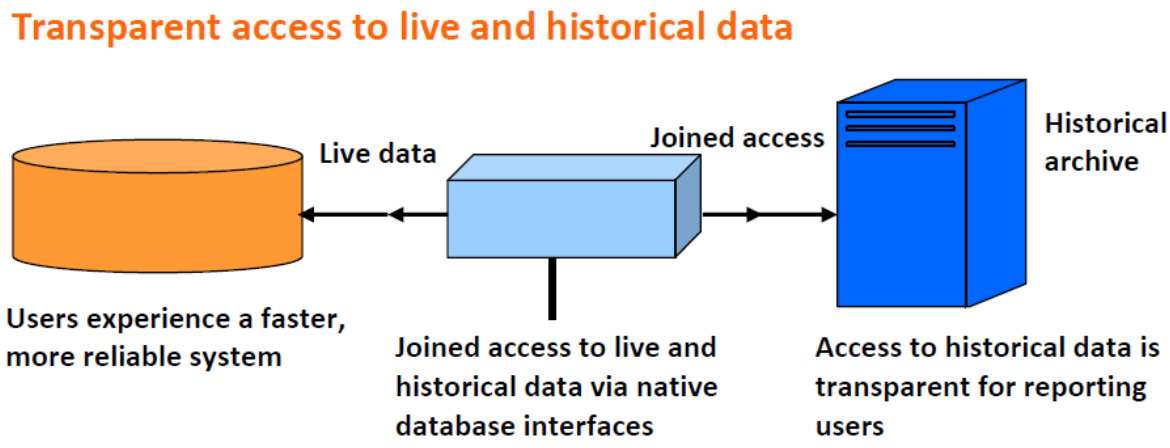
Produkten försöker klassificera datan i en databas utgående från hur viktig och relevant den är. Denna klassificering är mycket viktig för företagets ILM strategi eftersom det tillåter databaser att arkiveras samtidigt som referensintegriteten hålls intakt. (Grid-Tools 2010)

Dataarchive väljer ut datan som arkiveras utgående från tidigare aktivitet samt enligt relevanta bevaringsstrategier. Denna data flyttas sedan över i en ”History Schema”. Detta ”History Schema” kan antingen ligga på samma databasserver som produktions-databasen. Mest nytta får dock användaren ifall den inte gör det utan på en helt annan server. Nyttan får man av att systemkraven på produktionsserverns minskar i och med att datamängden minskar. Samtidigt sparar man också tid och lagringsutrymme på backup eftersom historisk data inte behöver sparas lika frekvent som aktiv data.



Figur 17. Databas uppdelningen i dataarchive (Grid-tools, 2010)

För att uppfylla juridiska krav är den arkiverade datan alltid möjlig att nå via en så kallad ”kombinerad vy”.
(Grid-tools, 2010)



Figur 18. Principen i dataarchive(Grid-Tools 2010)

För att underlätta användningen skapas en så kallad ”Joined Schema” som det ursprungliga programmet direkt kan använda. Detta underlättar också användningen till en stor grad eftersom användaren inte behöver lära sig ett nytt system.
Ifall man skulle behöva föra tillbaka arkiverad data in i produktionsdatabasen fungerar också detta.
(Grid-tools 2010)

Långtidsarkiveringskraven i dataarchive uppfylls enligt dokumentationen genom att man har möjlighet att spara den historiska datan i flat file format. Som sedan kan arkiveras enligt gällande standarder för text filer. Samma data kan sedan läsas in på nytt i system ifall det finns behov av detta senare.

6 AVSLUTNING

Eftersom det inte finns någon teknisk möjlighet att utföra arkivering och speciellt inte långtidsförvaring av material så kommer jämförelsen att göras helt utgående från det skriftliga materialet som finns tillgängligt.

6.1 Jämförelse och diskussion

För arkivering av databaser är det helt klart SIARD och RODA principerna som är de mest populära. Detta kan vi enkelt dra som slutsats eftersom dessa är de som valts som standarder inom stora projekt som exempelvis det sameuropeiska PLANETS projektet. Dessa två system är mycket liknade och bygger båda på att migrera de ursprungliga databasfilerna till versioner av XML filer. Därtill samlas korrekt metadata in och sparas för databasen automatiskt. Och sedan sparas alla dessa paket i långtidsarkiveringsdugliga filer direkt som uppfyller alla standarder som sätts på sådana. Det finns också automatisk loggning som lägger till data då en fil kontrolleras eller används.

De negativa egenskaperna för SIARD och RODA är också samma för båda dvs att databasen som arkiveras skall vara en inaktiv databas som inte uppdateras eller ändras mera. Detta kan vara svårt att uppnå speciellt då det kommer till företagsdatabaser.

Då man går in för att skriva egna arkiveringsprinciper enligt exempelvis Jack Olsons exempel från Database Archiving boken står man inför en del andra problem istället. Man måste själv samla in all den metadatan man vill ha och bestämma vad som är nödvändigt och vad som inte är det. Därtill sparas filerna inte automatiskt i långtidsarkiveringsdugliga paket utan detta måste man göra via något annat system exempelvis något som arkiverar textfiler. Loggningen av vad som händer i dessa filer under arkiveringen kan heller inte säkerställas så dessa filer kan aldrig riktigt uppfylla kraven för långtidsarkivering.

Det positiva med de självskrivna rutinerna är dock klara. Man kan ju här plocka bara delar ur en databas med speciella frågor och villkor och på så vis arkivera bara en del av databasen. Man får alltså en mycket större frihet över materialet.

För det kommersiella allternativet är fördelarna att systemet är enkelt att introducera i ett databassystem samt att systemet kan arkivera aktiva databaser eller delar av dessa. Nackdelen är dock att det inte utgår från dokumenten företaget ger hur långtidsbevaringsaspekten uppnås annat än att filerna kan sparas i flat file format.

Utgående från jämförelsen jag gjorde skulle jag vilja påstå att man välja bland dessa är det helt klart SIARD och RODA lösningarna som ger de bästa resultaten för långtidsarkivering.

6.2 Nästa steg

Nästa steg som borde tas är helt klart att man anmäler sig till RODA eller SIARD projekten för att ta del av verktygen. Eftersom databasarkivering överlag är ett relativt nytt problem kan det också snabbt komma flera olika allternativ på markande, speciellt kommersiella verktyg ämnade för en företagsmiljö. Ett annat intressant fortsatt vinkling kunde vara att se på hur man kan försäkra sig om att det arkiverade materialet hålls läsdugligt under en lång tid framöver.

KÄLLOR

Olson Jack. 2009. *Database Archiving: How to Keep Lots of Data for a Very Long Time*
ISBN: 978-0-12-374720-4 s.275

Stephens, Ryan; Plew, Ronald. 2000. *Database Design*
ISBN: 0-672-31758-3 s.527

CCSDS. 2002. 650.0-B-1 Reference Model for an Open Archival Information System (OAIS). Tillgänglig: http://public.ccsds.org/publications/archive/650x0b1.pdf
148 s. Hämtad 20.10.2010

CRL, The Center for Research Libraries; OCLC Online Computer Library Center, Inc. 2007. *Trustworthy Repositories Audit & Certification: Criteria and Checklist*
Tillgänglig: http://www.crl.edu/sites/default/files/attachments/pages/trac_0.pdf
Hämtad 25.10.2010

Bernstein, Amir. *Database Preservation: The international Challenge and the Swiss Solution*
Tillgänglig:
http://www.digitalpreservationeurope.eu/publications/briefs/database_preservation.pdf
Hämtad: 11.10.2011

Digital Preservation Europe. 2006. What is digital preservation?
http://www.digitalpreservationeurope.eu/what-is-digital-preservation/
Hämtad 13.10.2011

Stawowczyk Long, Andrew. 2009. *LONG-TERM PRESERVATION OF WEB ARCHIVES – EXPERIMENTING WITH EMULATION AND MIGRATION METHODOLOGIES*
Tillgänglig: http://netpreserve.org/publications/NLA_2009_IIPC_Report.pdf
Hämtad: 01.02.2012

Faria, Luis. 2011. RODA Repository of Authentic Digital Objects – Flyer.
Tillgänglig: http://redmine.keep.pt/attachments/24/or09_flyer.pdf
Hämtad: 03.03.2012

Freitas, Ricardo André Pereira; Ramalho, José Carlos. 2011 *Preservation of Relational Databases Significant Properties in the Preservation of Relational Databases*.
Tillgänglig:
http://repositorium.sdum.uminho.pt/bitstream/1822/13704/1/MSKE2011.pdf
Hämtad: 20.3.2012

Ramalho, José Carlos; Ferreira, Miguel; Faria, Luís; Castro, Rui. 2007. *Relational Database Preservation through XML modelling*,
Tillgänglig:
http://repositorium.sdum.uminho.pt/bitstream/1822/7120/1/EML2007-final.pdf
Hämtad: 28.3.2012

Factor, Michael; Henis, Ealan; Naor, Dalit; Rabinovici-Cohen, Simona; Reshef, Petra; Ronen, Shahar; Michetti, Giovanni; Guercio, Maria. 2009. *Authenticity and Provenance in Long Term Digital Preservation: Modeling and Implementation in Preservation Aware Storage*
Tillgänglig: http://www.usenix.org/event/tapp09/tech/full_papers/factor/factor.pdf
Hämtad: 26.10.2011

SIARD Format Description. 2009.
Tillgänglig:
http://www.bar.admin.ch/dienstleistungen/00823/00825/index.html?lang=en&download=NHzLpZeg7t,lnp6I0NTU042l2Z6ln1ad1IZn4Z2qZpnO2Yuq2Z6gpJCDdIR8fmym162epYbg2c_JjKbNoKSn6A--
Hämtad: 12.11.2011

Factsheet SIARD , 2010,
Tillgänglig:
http://www.bar.admin.ch/dienstleistungen/00823/00825/index.html?lang=en&download=NHzLpZeg7t,lnp6I0NTU042l2Z6ln1ad1IZn4Z2qZpnO2Yuq2Z6gpJCDdIR8fmym162epYbg2c_JjKbNoKSn6A--
Hämtad: 12.11.2011

Grid-Tools *Data Archive MANAGE LONG TERM DATA GROWTH AND DATABASE PERFORMANCE*. 2010.
Tillgänglig: http://www.grid-tools.com/download/Data_archive_technical_flyer.pdf
Hämtad: 01.02.2012

Grid-Tools.2010. *Data Archiving Strategies*
Tillgänglig: http://www.grid-tools.com/download/data_archiving_strategies.pdf
Hämtad: 01.02.2012

DuCharme, Bob. 2005. *25 years of database history (starting in 1955).*
Tillgänglig: http://www.snee.com/bobdc.blog/2005/12/25-years-of-database-history-s-1.html
Hämtad: 15.10.2011

O'Neil, Patrick; O'Neil, Elizabeth. 2001. Database--principles, Programming, and Performance.
Tillgänglig:
http://books.google.fi/books?id=UXh4qTpmO8QC&printsec=frontcover&dq=database&hl=en&ei=4VyTeDDCeOS4gavqa36BQ&sa=X&oi=book_result&ct=result&redir_esc=y#v=onepage&q&f=false
Hämtad: 11.10.2011

Descartes, Alligator; Bunce, Tim. 2000. *Programming the Perl DBI*
Tillgänglig: http://docstore.mik.ua/orelly/linux/dbi/ch03_01.htm
Hämtad 13.1.2012

Hand, Steve; Chandler, Jane. 1998. *INTRODUCTION TO OBJECT-ORIENTED DATABASES*
Tillgänglig:
http://www.odbms.org/download/005.01%20Chandler%20Introduction%20to%20Object-Oriented%20Databases%20September%201998.pdf
Hämtad 10.1.2012

Object database. 2012. Wikipedia
Tillgänglig: http://en.wikipedia.org/wiki/Object_database
Hämtad 10.1.2012

SAOL.2006. *Svenska Akademiens ordlista.*13:e upplagan. Norstedts Akademiska Förlag. 1130 s.

BILAGOR