

Anvisning i aflevering af bitemporale data

Indhold

Anvisning i aflevering af bitemporale data	1
Baggrund	2
Hvad er bitemporale data?	2
Format til aflevering af bitemporale data	3
SQL:1999-delen:	4
SQL:2011-delen:.....	4
Bilag 1: Eksempel på SQL:2011 DLL	5
Forklaring til eksempel på SQL:2011 DDL	6
Referencer	7

Baggrund

Aflevering af data fra it-systemer til et offentligt arkiv er baseret på aflevering af en arkiveringsversion i en relationel databasestruktur. Det indebærer, at sammenhænge mellem tabeller angives som fremmednøgler, der refererer primærnøgler. De gældende regler er fastlagt i bekendtgørelse om arkiveringsversioner, der er baseret på bevaringsformatet SIARD 1.0 og anvender et uddrag af standarden SQL:1999.

Temporale data er opmærket med tidsrum, der indgår i primær- og fremmednøgler til at identificere unikke rækker i én tabel og referere unikke rækker i andre tabeller, og sammenhænge mellem tabeller afhænger derfor af en fortolkning af disse tidsrum. SQL:1999 understøtter imidlertid ikke angivelse og fortolkning af primær- og fremmednøgler, hvori tidsrum indgår. Derfor er temporale data hidtil blevet omdannet og håndteret efter de gældende regler. Unitemporale data skal fortsat håndteres derefter, mens bitemporale data har vist sig noget mere omstændelige at håndtere udelukkende med anvendelse af SQL:1999.

For at undgå større ændringer af datamodel og data i forbindelse med aflevering af en arkiveringsversion, anviser rigsarkivaren med henvisning til § 3, stk. 2 i bekendtgørelse om arkiveringsversioner, hvordan bitemporale data efter nærmere aftale med det modtagende arkiv kan bevares i en fremtidssikret struktur.

Hvad er bitemporale data?

Bitemporale data er opmærket med en virkningstid og en registreringstid og understøtter dermed såkaldt bitemporalitet eller dobbelthistorik, hvormed et forvaltningsobjekts tilstand kan forstås som mængden af egenskaber, der karakteriserer det i en afgrænset del af tiden angivet i hver af de to dimensioner af tid.

Virkningstid – også kaldet valid time eller applikationstid – er tiden, hvor et forvaltningsobjekts egenskaber svarer til forholdet i virkeligheden. Det handler om, hvad forvaltningen ved og kunne f.eks. være en given persons efternavn i en bestemt periode. Hvis forvaltningen ved, at personen har forskellige efternavne i forskellige perioder, kan der registreres to efternavne med hver deres virkningstid, hvormed virkningstiden understøtter muligheden for at kende til, hvornår noget i den virkelige verden har ændret sig.

Registreringstid – også kaldet transaction time eller systemtid – er tiden, hvor et forvaltningsobjekts egenskaber er kendt i et bestemt system. Det handler om, hvornår forvaltningen ved noget og kunne f.eks. være registreringen af en given persons efternavn i en bestemt periode i et bestemt system. Hvis forvaltningen bliver opmærksom på, at personen havde et andet efternavn i en periode, end man tidligere vidste, kan den eller de tidligere registrering(er) afsluttes med det tidspunkt, som forvaltningen opdaterer personens efternavn i systemet, og der kan registreres et eller flere efternavne i en eller flere perioder i stedet. I en database, der ikke tager højde for registreringstid, ville man redigere de eksisterende registreringer og altid have én version af forvaltningsobjektet – den nuværende. Med registreringstid understøtter man derimod flere

versioner af forvaltningsobjektet som registreret i et bestemt system, således at man til et hvilket som helst tidspunkt kan kende de aktuelle registreringer i et bestemt system, hvormed registreringstid understøtter muligheden for tilbagerulning af data ved at bibeholde et fyldestgørende revisionsspor.

I eksemplet nedenfor blev personen med Id 32167 født med efternavnet Jacobsen den 22. januar 1992. Det blev dog først registreret en uge efter fødslen. 16. juli 2018 blev personen gift og tog sin ægtefælles efternavn. Det blev dog først registreret tre uger efter brylluppet – markeret med kursiveret skrift:

ID	Efternavn	VirkningFra	VirkningTil	RegistreringFra	RegistreringTil
32167	Jacobsen	1992-01-22	9999-12-31	1992-01-29	2018-08-06
32167	Jacobsen	1992-01-22	2018-07-16	2018-08-06	9999-12-31
32167	Knudsen	2018-07-16	9999-12-31	2018-08-06	9999-12-31

Som det fremgår, slettes eller overskrives rækker ikke i et system med dobbelthistorik. I stedet registreres, hvad personens efternavn er og har været, og hvornår det var kendt i systemet. Dermed kan man udbede sig personens efternavnehistorik den 23. april 2019 og få at vide, at denne hed Jacobsen fra 22. januar 1992 til 16. juli 2018, hvorefter denne har heddet Knudsen. Man kan også finde ud af, hvad der 1. august 2018 var registreret om personen, der på det tidspunkt og indtil 6. august i forvaltningens øjne var hr. Jacobsen.

Format til aflevering af bitemporale data

Rigsarkivet dispenserer fra gældende regler fastlagt i bekendtgørelse om arkiveringsversioner, eftersom bitemporale data kan afleveres som en arkiveringsversion i en relationel databasestruktur uden angivelse af temporale fremmednøgler. Som supplement dertil skal temporale primær- og fremmednøgler angives særskilt i tableIndex.xml med anvendelse af SQL:2011-standarden som

beskrevet ved ISO/IEC TR 19075-2. Det medfører, at arkiveringsversioner af it-systemer med bitemporale data indeholder følgende elementer:

SQL:1999-delen:

- Data afleveres som en arkiveringsversion i en relationel databasestruktur i overensstemmelse med bekendtgørelse om arkiveringsversioner. Primær- og fremmednøgler, hvori tidsrum indgår, skal ikke angives i `tableIndex.xml`. Der skal i stedet angives ikke-temporale primærnøgler til alle tabeller – f.eks. ved indsættelse af et fortløbende nummer – og alle data skal fortsat afleveres.
- Arkiveringsversionens SQL:1999-databasestruktur dokumenteres på sædvanlig vis i kontekst- dokumentationen. Databasestrukturen dokumenteres i den tekniske dokumentation af arkiverings- versionen, herunder med E/R-diagram, tabel- og feltbeskrivelser og konverteringsbeskrivelse(r).

SQL:2011-delen:

- Myndigheden udarbejder en datadefinition, der viser, hvordan arkiveringsversionens database- struktur ville se ud, hvis den anvendte SQL:2011-standarden med angivelse af PERIODs og temporale primær- og fremmednøgler. En datadefinition (DDL) er et databaseskema bestående af SQL-sætninger, der angiver databasestrukturen med tabeller og felter, nøgler og øvrige constraints. DDL'en indsættes som en [XML Processing Instruction](#) ved navn `<?SQL2011 DDL ?>` efter det afsluttende rodelement nederst i `tableIndex.xml`. Se desuden Bilag 1: Eksempel på SQL:2011 DDL.
- Myndigheden konverterer DDL'en til et TIFF-billede, der vedlægges kontekstdokumentation jf. de gældende regler i bekendtgørelse om arkiveringsversioner pkt. 4.E og pkt. 6.B.
- Myndigheden udarbejder et E/R-diagram, der viser databasestrukturen defineret i SQL:2011 DDL med primær- og fremmednøgler og tabelreferencer, der ikke findes i E/R-diagrammet for SQL:1999.
- Myndigheden udarbejder SQL-forespørgsler, der viser sammenhænge mellem tabeller angivet med anvendelse af SQL:2011. Forespørgslerne skal udformes i overensstemmelse med SQL:2011- standarden som eksemplificeret i afsnittene 4.1.8 og 4.2.4 samt 4.3 i rapporten ISO/IEC TR 19075-2 og indsættes i `tableIndex.xml` jf. de gældende regler i bekendtgørelse om arkiveringsversioner pkt. 6.D og figur 6.3, 7. Elementet `<name>` skal begynde med "AV_DDL_".
- Det modtagende arkiv skriver i afleveringsbestemmelsen, at arkiveringsversionen indeholder bitemporale data og er afleveret efter nærværende bestemmelse. Afleveringsbestemmelsen udarbejdes af det modtagende arkiv og vedlægges kontekstdokumentation til arkiveringsversionen.

Bilag 1: Eksempel på SQL:2011 DLL

I eksemplet herunder er DDL-syntaksen, der kun anvender SQL:2011-standarden, markeret med **blå skrift**:

```

CREATE TABLE Bygning (
    RowKey INTEGER,

    Id VARCHAR(50),

    VirkningFra TIMESTAMP NOT NULL,
    VirkningTil TIMESTAMP NOT NULL,

    PERIOD FOR BygningVirkning (VirkningFra, VirkningTil),
    RegistreringFra TIMESTAMP GENERATED ALWAYS AS ROW START,
    RegistreringTil TIMESTAMP GENERATED ALWAYS AS ROW END,

    PERIOD FOR SYSTEM_TIME (RegistreringFra, RegistreringTil),

    ...

) WITH SYSTEM VERSIONING;

ALTER TABLE Bygning ADD PRIMARY KEY (Id, BygningVirkning WITHOUT OVERLAPS);

CREATE TABLE Opgang (
    RowKey INTEGER,

    Id VARCHAR(50),

    VirkningFra TIMESTAMP NOT NULL,
    VirkningTil TIMESTAMP NOT NULL,

    PERIOD FOR OpgangVirkning (VirkningFra, VirkningTil),
    RegistreringFra TIMESTAMP GENERATED ALWAYS AS ROW START,
    RegistreringTil TIMESTAMP GENERATED ALWAYS AS ROW END,

    PERIOD FOR SYSTEM_TIME (RegistreringFra, RegistreringTil),
    Bygning_id VARCHAR(50),

    ...

) WITH SYSTEM VERSIONING;

ALTER TABLE Opgang ADD PRIMARY KEY (Id, OpgangVirkning WITHOUT OVERLAPS);

```

```
ALTER TABLE Opgang ADD FOREIGN KEY (Bygning_id, PERIOD
OpgangVirkning) REFERENCES Bygning (Id, PERIOD BygningVirkning);
```

Forklaring til eksempel på SQL:2011 DDL

Hjørnестenen i understøttelsen af bitemporale data i SQL:2011 er muligheden for at angive tidsrum og associere dem med rækker i en tabel. Et tidsrum er et interval med en start- og sluttid. Ved at definere to kolonner som en **PERIOD** kan et SQL:2011-kompatibelt RDBMS sikre referentiel integritet ved at fremmednøglen tidsrum er indeholdt i primærnøglen tidsrum. **WITHOUT OVERLAPS** modvirker overlappende virkningstidsrum for registreringer med den samme identifikator, som indgår i primærnøglen, når man foretager nye registreringer for denne identifikator, og starttidspunktet – men ikke sluttidspunktet – medtages i tidsrummet i et lukket-åbent-interval, som SQL:2011-standarden anvender.

I et RDBMS, der har implementeret SQL:2011-standarden for temporale data vil en **SELECT** kun resultere i rækker med aktuelle værdier, dvs. hvor registreringstiden er uafsluttet, hvorimod rækker med uaktuelle værdier – f.eks. den første række i eksemplet med efternavne – skal fremsøges med **FOR SYSTEM_TIME**.

Bilag 1 er et eksempel på dele af den DDL, der anvendes til at angive en bitemporal databasestruktur for de eksisterende data med anvendelse af SQL:2011-standarden. Først angives en **CREATE TABLE**, der med syntaksen **PERIOD FOR** definerer de to kolonnepar som tidsrum. **PERIOD FOR** er en period definition, der har den fordel, at den fungerer sammen med eksisterende datatyper – f.eks. datotyper. Virkningstiden er angivet som **BygningVirkning**, mens registreringstidens periode angives som **SYSTEM_TIME**.

WITH SYSTEM VERSIONING angiver, at tabellen er en systemversioneret tabel, der bevarer en præcis historik over alle ændringer i data. Det vil sige, at tabellen indeholder kolonner med registreringstider, hvor it-systemet automatisk opdaterer datoværdierne, hver gang rækker bliver ændret eller slettet i tabellen og samtidig bevarer slettede og redigerede rækker som uaktuelle – afsluttede – rækker i databasen.

ALTER TABLE angiver de temporale primær- og fremmednøgler. I tabellen **Bygning** er primærnøglen sammensat af forretningsnøglen **Id** og virkningstiden **BygningVirkning**. I tabellen **Opgang** er fremmednøglen sammensat af **Bygning_id**, der referer forretningsnøglen **Id** i tabellen **Bygning**, og virkningstiden **OpgangVirkning** for den tabel **Opgang**, der refereres fra.

Referencer

[Bekendtgørelse om arkiveringsversioner] Kulturministeriet. [BEK nr. 128 af 12. februar 2020](#)

[Bitemporalitet] OIO Arkitekturguiden. [Bitemporalitet – Proof of concept](#). 19. September 2014.

[ISO/IEC TR 19075-2] ISO/IEC JTC 1/SC 32. Data management and interchange. [ISO/IEC TR 19075-2. Information technology – Database languages – SQL Technical Reports – Part 2: SQL Support for Time-Related Information](#). Juli 2015.

[SIARD 1.0] E-Government Standards. eCH-0165. [Software-Independent Archival of Relational Databases](#).

[SQL:1999] ISO/IEC JTC 1/SC 32. Data management and interchange. ISO/IEC 9075:1999. Information technology – Database languages – SQL. December 1999

[SQL:2011] ISO/IEC JTC 1/SC 32. Data management and interchange. ISO/IEC 9075:2011. Information technology – Database languages – SQL. December 2011