

**IBM Information**  
>>> On Demand

2006



# Design your application for DB2 for z/OS V9

*Peter Hartmann / [peterhar@de.ibm.com](mailto:peterhar@de.ibm.com)*

*Session 1855A*

*20. October 2006*



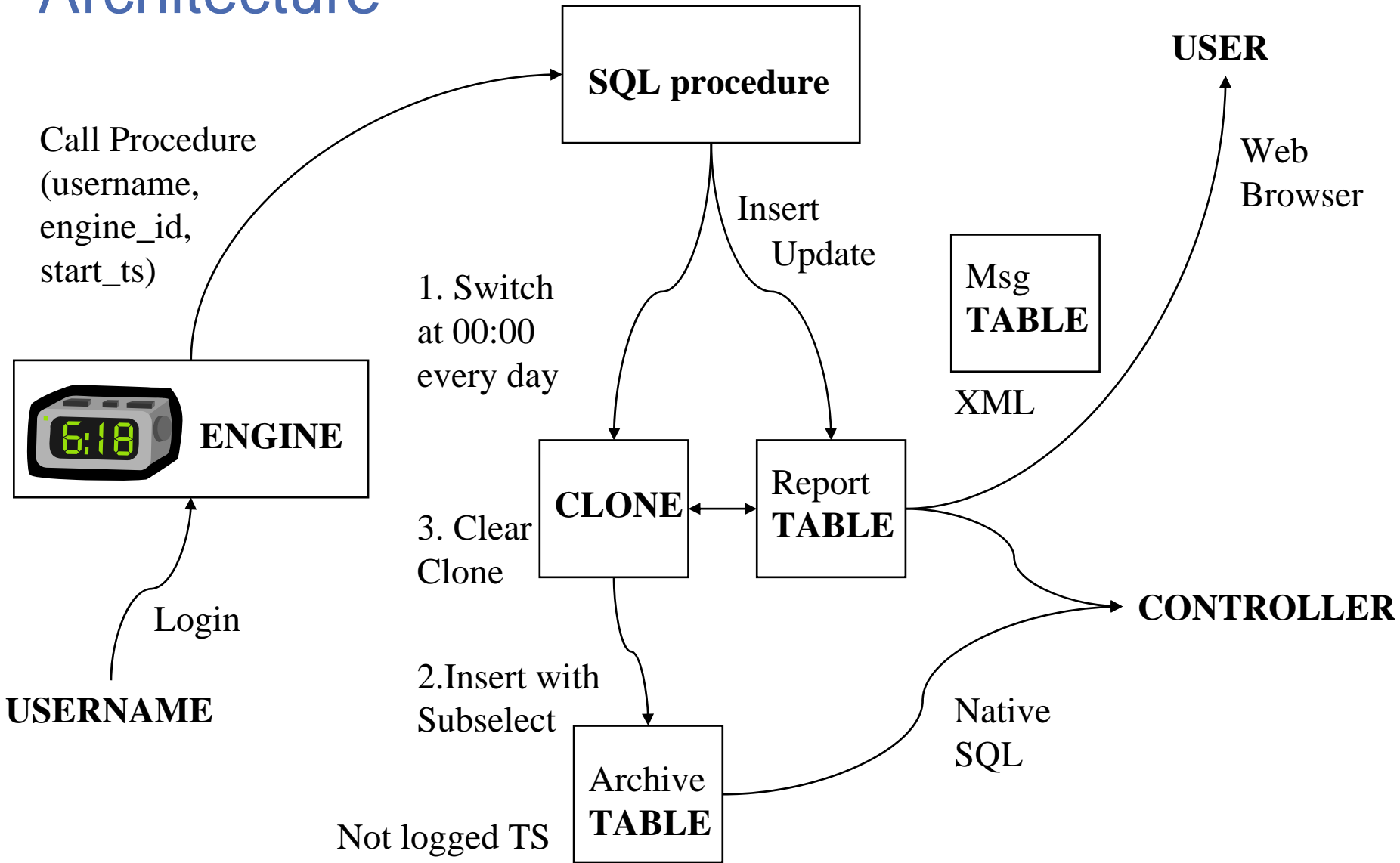
**ON** DEMAND BUSINESS™

# Contents

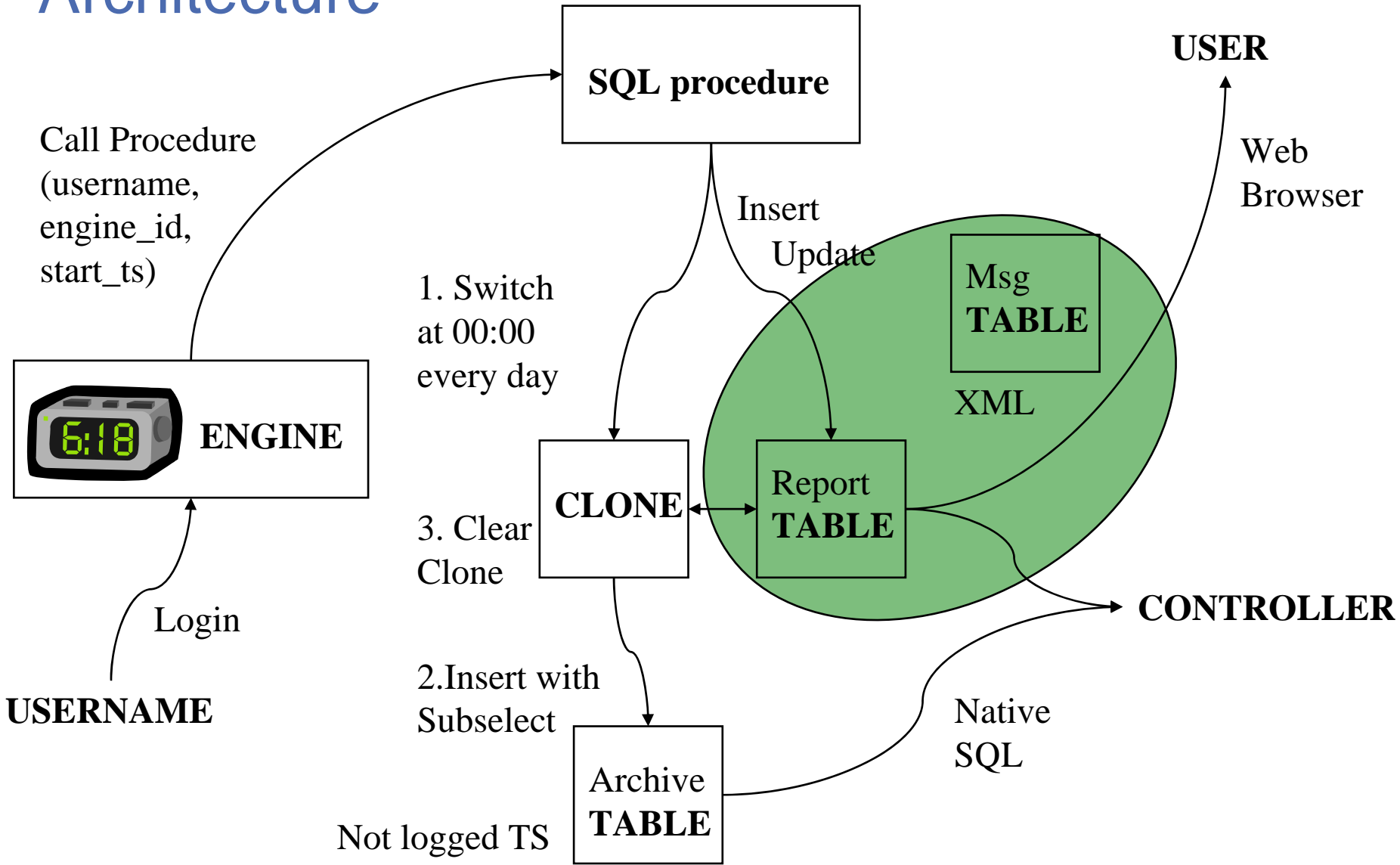
- Architecture
- New datatypes
- SQL enhancements
- Administration
- Query enhancements
- Summary



# Architecture

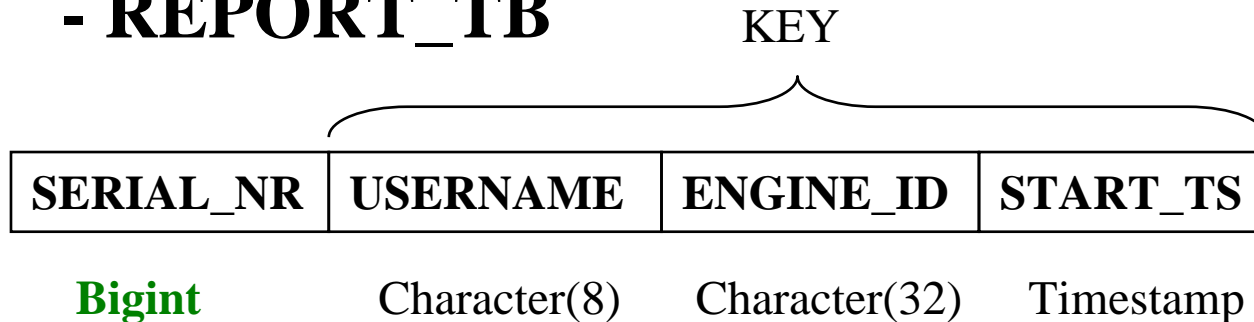


# Architecture



# Table definitions

## - REPORT\_TB



Other attributes:

-Locksize Row



```
CASE
WHEN COUNTER BETWEEN xx AND yy THEN
  (SELECT MSG_TEXT FROM MSG_TB WHERE MSG_ID=1)
ELSE ...
END
```

## - MSG\_TB



Integer

**XML**



# Table definitions

```
CREATE TABLE REPORT_TB
(
SERIAL_NR          BIGINT NOT NULL ,
USERNAME          CHAR(8)          ,
ENGINE_ID         CHAR(32) NOT NULL ,
START_TS          TIMESTAMP NOT NULL WITH DEFAULT,
LAST_REPORT_TS    TIMESTAMP NOT NULL
GENERATED ALWAYS
FOR EACH ROW ON UPDATE AS ROW CHANGE TIMESTAMP,
COUNTER           DECFLOAT WITH DEFAULT 1
)
```

```
CREATE TABLE MSG_TB
(
MSG_ID            INTEGER ,
MSG_TEXT          XML
)
```



# New datatypes

*TAKE BACK CONTROL*



# Bigint

- Double word
- Range
  - Smallint  $2^{15}-1 = 32767$
  - Integer  $2^{31}-1 = 2147483647$
  - Bigint  $2^{63}-1 = 9223372036854775807$   
to  $-9223372036854775808$
- Compatible with all other numeric types
- Also new build-in function Bigint





# Bigint

- SERIAL\_NR is addressed by sequence
  - Definition

```
CREATE SEQUENCE REPORT_SEQUENCE  
AS BIGINT  
START WITH 1 INCREMENT BY 1 NO CYCLE
```

- Insert

```
INSERT INTO REPORT_TB  
( SERIAL_NR, ... )  
VALUES  
( NEXT VALUE FOR REPORT_SEQUENCE , ... )
```



# Change Timestamp column

- Value is generated
  - By insert of row
  - By update on any column of the row
- Can be defined with „hidden“
  - Select \* does not return this column
- Can be used for „optimistic locking“
  - Test in application whether the underlying row has been updated by another transaction since the last Select
  - New expressions
    - Row Change Timestamp
    - Row Change Token





# Decfloat

- INF (=infinity) and NaN (=not a number)

```
CREATE ... (COL_DECFLOAT DECFLOAT(34) )
...
INSERT INTO ... VALUES (INF)
...
SELECT CHAR(COL_DECFLOAT - INF) AS RESULT FROM ...
...
RESULT
-----
NaN
```

- Some rules

- Infinity + 1           ⇒ Infinity
- Inf + Inf             ⇒ Infinity
- NaN + 1               ⇒ NaN
- NaN + Infinity       ⇒ NaN
- 1 - Infinity         ⇒ -Infinity
- ...



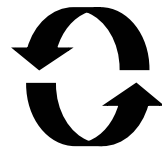
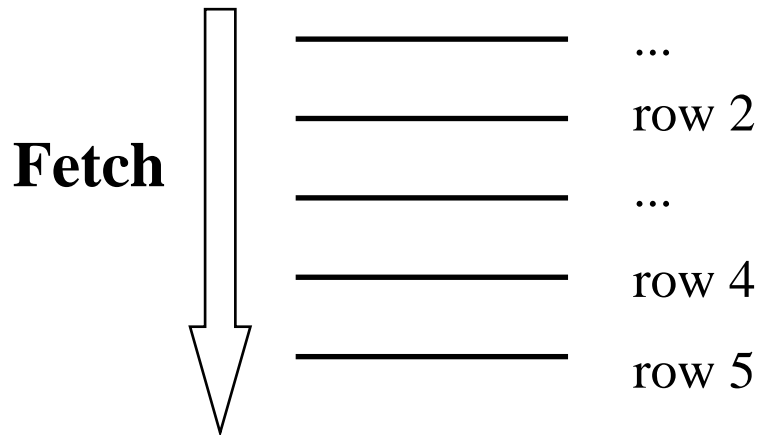
# XML

- Well-formed XML document
  - Start tag: <...>
  - End tag: </...>
  - Parsing is done by z/OS XML parser (OA16303)
- XML schema support
  - Stored procedures
- Indexes can be defined
- Relational language is enhanced for XML documents



# XML

- In-place update not possible
- Fetch „with continue“ option
  - Also for LOB columns
  - Normal fetch versus fetch continue



## Fetch with continue

Truncation is indicated by  
`Sqlwarn1=W Sqlcode+20141`

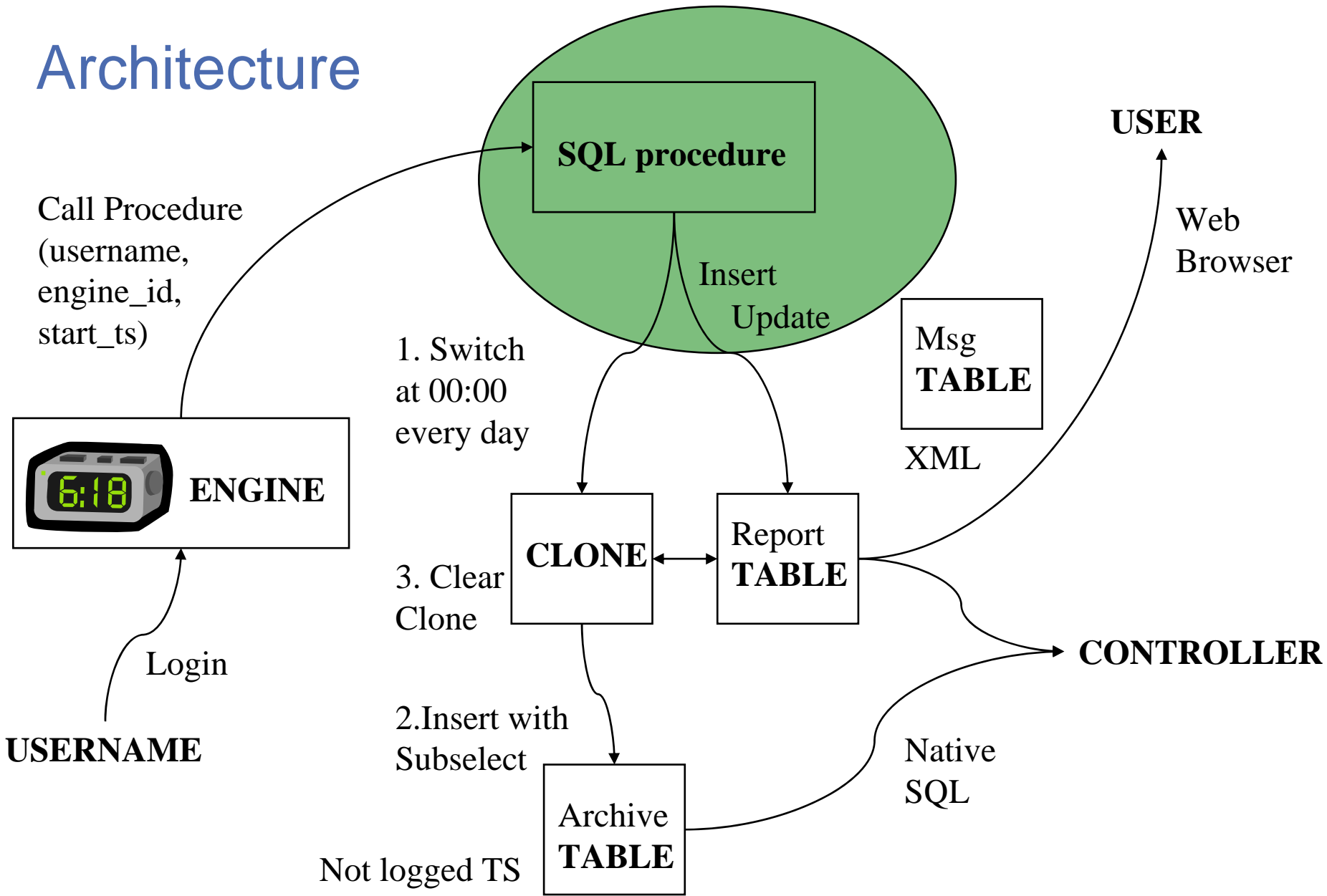


# XML

- Data is stored in own pagesets (created implicitly by DB2)
  - XML tablespace and nodeid index
  - Docid index on base table (row generated implicitly: DB2\_GENERATED\_DOCID\_FOR\_XML)
  - Attributes (e.g. Stogroup, ... ) are inherited from base tablespace
  
- No design limit in size (2 GB in architecture)



# Architecture





# SQL enhancements

*TAKE BACK* **CONTROL**



# SQL procedures

- Runtime
  - Integrated into the engine (DBM1)
  - Debugging in WLM application environment
- Versioning
  - Several versions can exist in parallel
  - Called version of a procedure can be
    - Active one
      - New flag in Sysibm.Sysroutines
    - „Current Routine Version“ register



# SQL procedures

```
CREATE PROCEDURE
REPORT_SQL_PROCEDURE
(IN P1 CHAR(8),
 IN P2 CHAR(32),
 IN P3 TIMESTAMP)
VERSION V001
LANGUAGE SQL
MODIFIES SQL DATA
DISABLE DEBUG MODE
MERGE INTO REPORT_TB T1
...
SET T1.COUNTER=T1.COUNTER+1
...
```

**Active version**

```
ALTER PROCEDURE
REPORT_SQL_PROCEDURE
ADD VERSION V002
(IN P1 CHAR(8),
 IN P2 CHAR(32),
 IN P3 TIMESTAMP)
MODIFIES SQL DATA
DISABLE DEBUG MODE
MERGE INTO REPORT_TB T1
...
SET T1.COUNTER=T1.COUNTER+2
...
```

**Active version**

```
ALTER PROCEDURE REPORT_SQL_PROCEDURE
ACTIVATE VERSION V002
```



# SQL procedures

- Other Alter options are
  - Drop Version
    - Drop procedure deletes all versions
  - Regenerate Version (e.g. for DB2 maintenance)
  - Replace Version
- SQL text is stored in CLOB column in Sysibm.Sysroutines



# Merge

## REPORT\_TB

KEY	Counter
aaa	1
bbb	1
ddd	2

+

:hv array=**M1**

KEY
bbb
eee

=

## REPORT\_TB

KEY	Counter
aaa	1
bbb	<b>2</b>
ddd	2
<b>eee</b>	<b>1</b>

Merge into **REPORT\_TB T1**

Using (Values (:hv array) For 1 Rows) As **M1**(KEY)

On (**T1**.KEY=**M1**.KEY)

When Not Matched Then Insert (KEY) Values (**M1**.KEY)

When Matched Then Update **T1**.Counter=**T1**.Counter+1

Not Atomic Continue On Sqlexception



# Merge

- Combined Insert and Update
- Multi row operation
  - Not Atomic Continue on SQLException
  - Error diagnose by „Get Diagnostics“ !
- Keyword „Default“ can be used
  - Already in Insert
  - Also new in Update

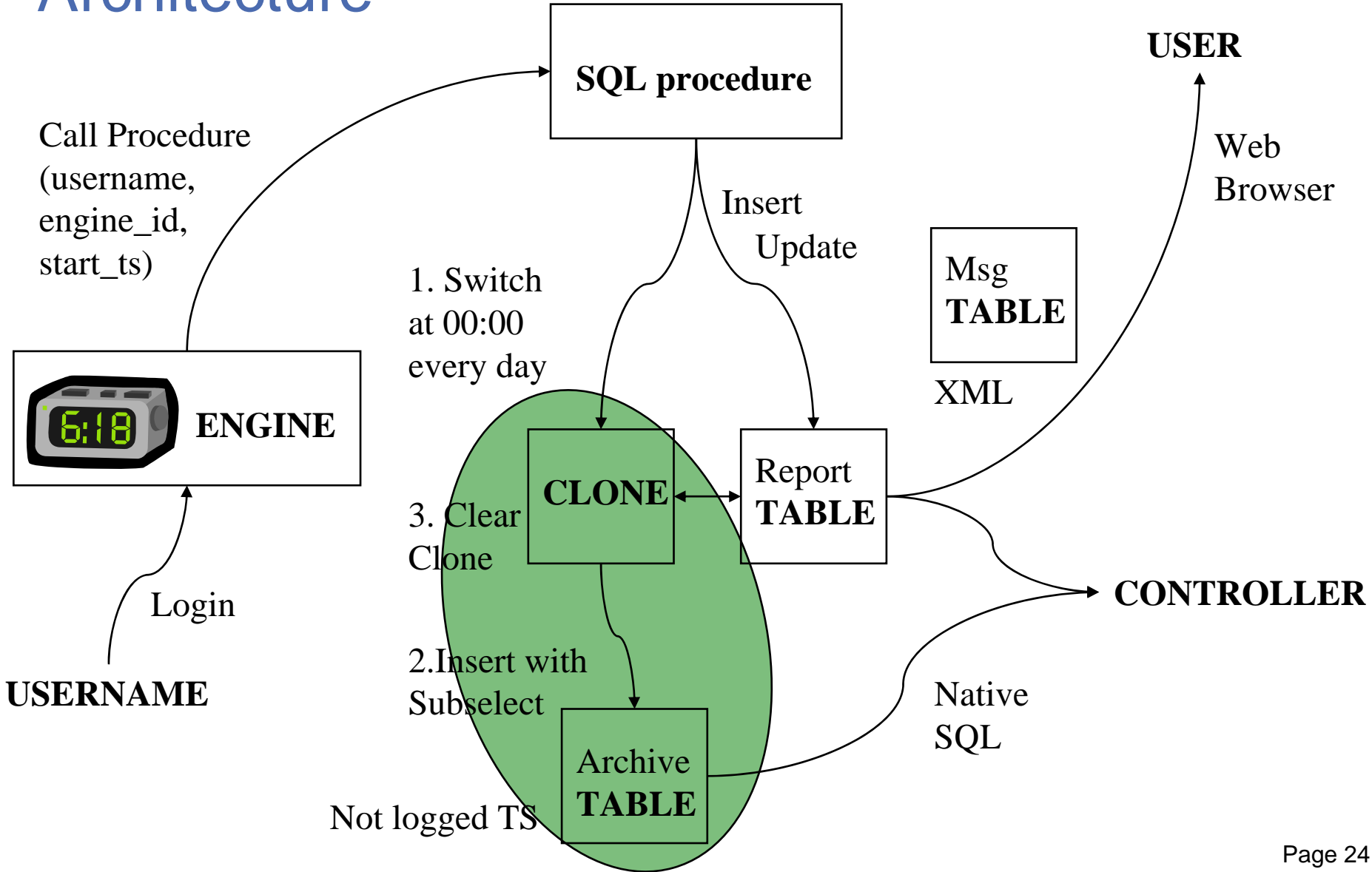


# Merge

- V8: Select from Insert
  - E.g. Select ... From **Final Table** (Insert Into ... Values ...)
- New keywords
  - **Final table**      Insert, Merge, Update
  - **Old table**        Delete, Update
- E.g. Select ... From **Old Table** (Delete From ... Where ...)



# Architecture





# Administration

*TAKE BACK* **CONTROL**



# Clone table

**MERGE**



**REPORT\_TB**

**Create Table  
REPORT\_TB**

**Alter Table REPORT\_TB Add  
Clone REPORT\_TB\_\_CLONE**

**Exchange Data Between  
Table REPORT\_TB And  
REPORT\_TB\_\_CLONE**

**Pagesets**

**...I0001...**

obid x'0...'

**Pagesets**

**...I0002...**

obid x'8...'

**1. Unload REPORT\_TB\_\_CLONE  
by Insert with Subselect**

**2. Clear REPORT\_TB\_\_CLONE  
by **Truncate****



# Clone table

- Only one table can be in the tablespace
  - Universal tablespace !
- Alter statement
  - on table is restricted
  - for Drop Clone
- New catalog field
  - **Instance** for I0001 or I0002



# Clone table

- Utility and command support by keyword **Clone**
  - E.g. Recover ... **Clone**, but not Runstats
  - Start / Stop Database ... **Clone**

- Display database

NAME	TYPE	PART	STATUS
-----	-----	-----	-----
SZI30S	<b>TSB1</b>	0001	RW
SZI30S	<b>TSC2</b>	0001	RW
SZI30X	<b>IXB1</b>	L*	RW
SZI30X	<b>IXC2</b>	L*	RW



# Universal table space

- Segmented and partitioned tablespace can be combined
  - Numparts and Segsize: Range partitioned
  - Maxpartitions and Segsize: Size partitioned
    - „Partition by growth“



# Partition by growth

- Create allocates only the first partition, others are allocated by space usage
  - Up to Maxpartitions (new catalog field)
- Create table
  - Partition By Size (if implicitly created)
  - Append (also by Alter)
    - Fast insert processing



# Not logged tablespace

- Archive table
  - Resides in a **partition by growth** and **not logged** tablespace
  - Filled by Insert with subselect
  - Image copy after the insert to establish a recovery point

```
CREATE TABLESPACE ARCHIVTS IN REPORTDB  
SEGSIZE 4 MAXPARTITIONS 500  
USING STOGROUP DB200G PRIQTY 1000 SECQTY 1000  
BUFFERPOOL BP4  
NOT LOGGED
```



# Not logged tablespace

- During Create or Alter
- LOBs, XMLs and indexes are dependent
- Utility Log option
  - No logging wins (either utility or tablespace)
- No rollback or backout possible otherwise RECP (tablespace) or RBDP (index)
  - Also in internal processing (-803)
- Recovery is still possible to „recoverable points“
  - e.g. Image Copy Shrlevel Reference



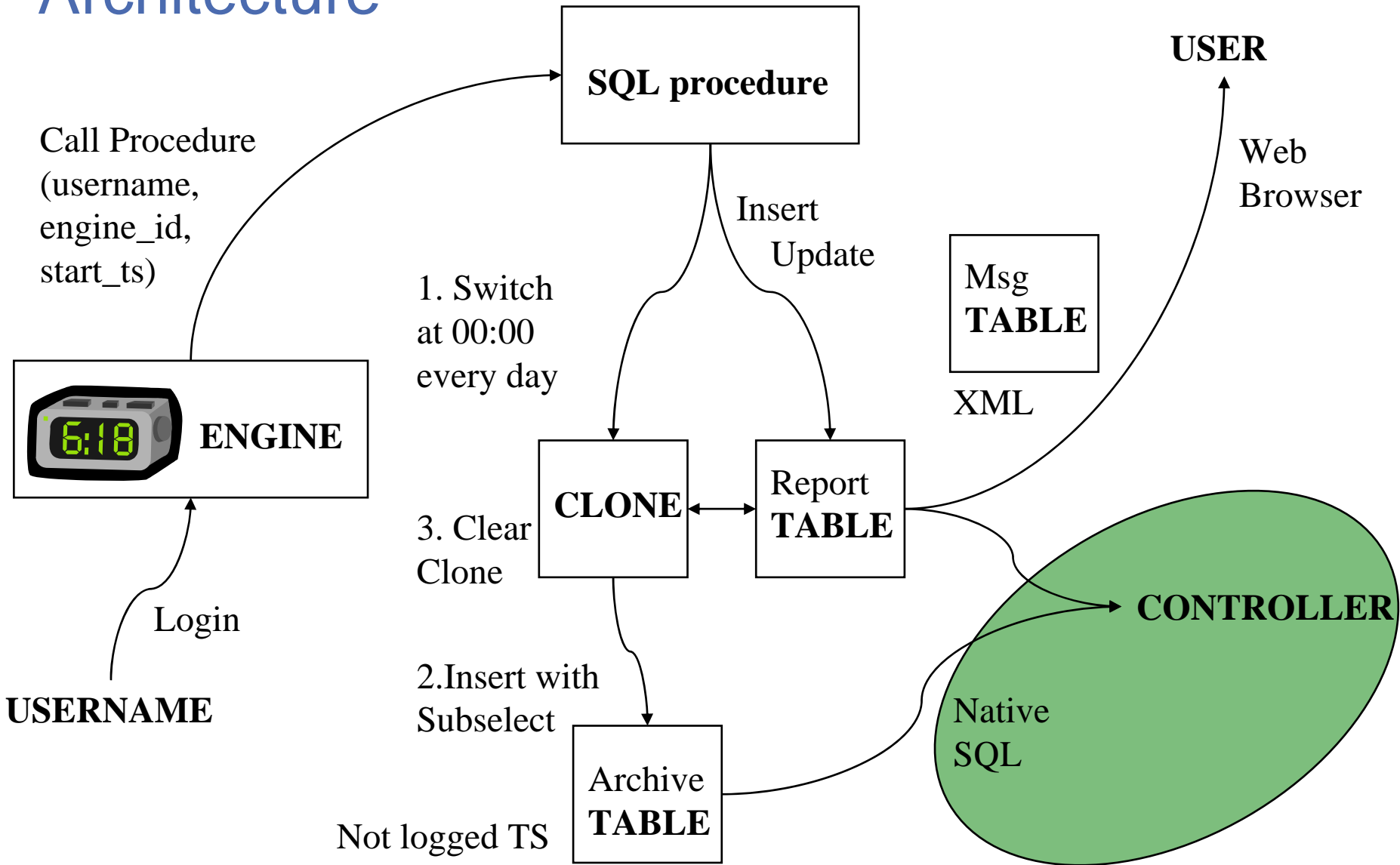


# Truncate table ~ Mass delete

- All space types are supported
  - Fast path logic only for segmented or universal tablespaces
  - No fast path if
    - Change data capture
    - Multi level security
    - Validproc
- Enhancements are
  - Immediate: No rollback
  - Ignore Delete Triggers



# Architecture



# Query enhancements

*TAKE BACK* **CONTROL**



# Order by enhancements

- Fetch first and Order by

```
SELECT SERIAL_NR, USERNAME, ENGINE_ID,  
START_TS, LAST_REPORT_TS,  
CHAR(COUNTER)  
FROM REPORT_TB  
ORDER BY COUNTER DESC  
FETCH FIRST 3 ROW ONLY
```

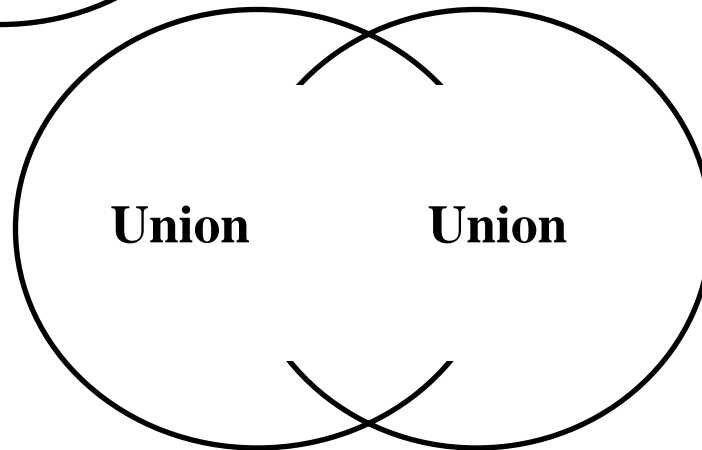
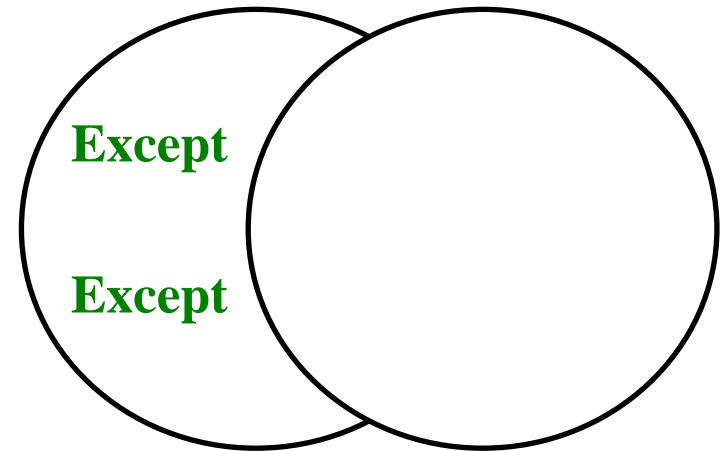
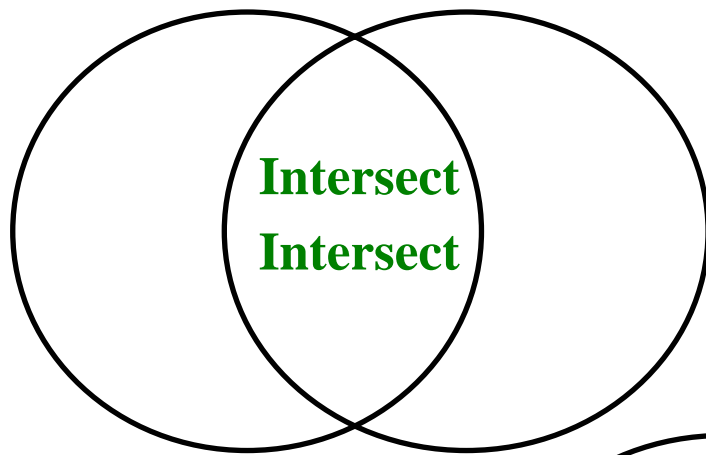
- Cultural sort (OA16037)

```
SELECT USERNAME  
FROM REPORT_TB  
ORDER BY COLLATION_KEY(USERNAME, 'DE_DE')
```



# Except and Intersect

- Enhancement to Union



# Except and Intersect

- All who worked **yesterday** and **today**

```
SELECT USERNAME, ENGINE_ID FROM ARCHIVE_TB
WHERE
EXTRACT(YEAR FROM START_TS) =
EXTRACT(YEAR FROM CURRENT_TIMESTAMP-1 DAY)
AND
EXTRACT(DAY FROM START_TS) =
EXTRACT(DAY FROM CURRENT_TIMESTAMP - 1 DAY)
INTERSECT
SELECT USERNAME, ENGINE_ID FROM REPORT_TB
WHERE
EXTRACT(DAY FROM START_TS) =
EXTRACT(DAY FROM CURRENT_TIMESTAMP)
AND
EXTRACT(YEAR FROM START_TS) =
EXTRACT(YEAR FROM CURRENT_TIMESTAMP)
```



# Build-in functions

- Extract
  - Date (year, month, day) or Time (hour, minute, second)
  - E.g. `Select Extract(Day From Current Timestamp) From Sysibm.Sysdummy1`
  
- Varchar\_Format
  - E.g. `Select Varchar_Format (Current Timestamp, 'YYYY-MM-DD HH24:MI:SS' ) From Sysibm.Sysdummy1` results in 2006-04-24 09:00:05



# Index on expression

```
INSERT INTO REPORT_TB  
(... , USERNAME , ... )  
VALUES  
(... , ` Peter ` , ...
```

```
INSERT INTO REPORT_TB  
(... , USERNAME , ... )  
VALUES  
(... , ` pETER ` , ...
```

```
INSERT INTO REPORT_TB  
(... , USERNAME , ... )  
VALUES  
(... , ` PeTEr ` , ...
```





# Index on expression

- Access path problem
  - Expression is not indexable and stage2 (tablespace scan):  
E.g. Select ... From ... Where  
Upper(Username)=,PETER'
- Example
  - Create Index ... On ...  
(Upper(Username,'De\_De') Asc )



# Index on expression

- Query must match the expression
  - E.g. Upper(Username) || Engine\_ID is not supported
- Enforcement during runtime, e.g.
  - Uniqueness against the index entry, not the table entry
  - Expression is 10 / Column1 and Column1 could be zero
- New catalog table: Syskeytargets
  - Describes the expression
  - Has statistics (Runstats)



# Skip locked data

- Applicable for incompatible locks
- Cursor and read stability (CS, RS)
- Only on row and page level
- No indication that rows are skipped



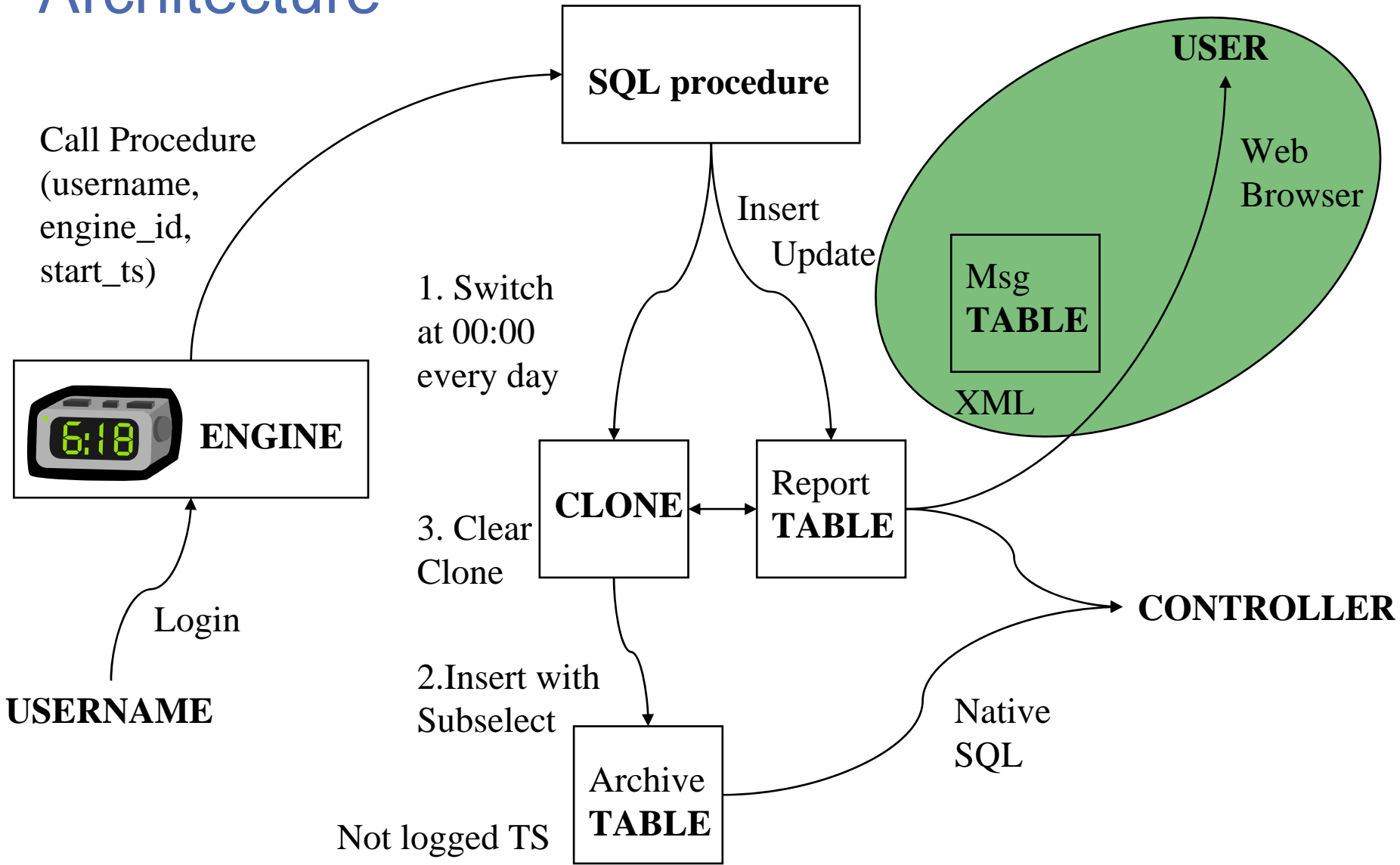
# Skip locked data

- Can be specified in
  - Searched Delete and Update
  - Select
  - Unload utility

```
UPDATE REPORT_TB  
SET START_TS = DEFAULT  
WHERE ENGINE_ID = '1111-1111-1111'  
SKIP LOCKED DATA
```



# Architecture



# XML table

```
INSERT INTO MSG_TB  
(MSG_ID, MSG_TEXT)  
VALUES  
( 1, „<DESCRIPTION>THIS IS THE REPORT DESCRIPTION 1</DESCRIPTION>“ )
```

```
INSERT INTO MSG_TB  
(MSG_ID, MSG_TEXT)  
VALUES  
( 1, „<DESCRIPTION>THIS IS THE REPORT DESCRIPTION 2</DESCRIPTION>“ )
```



# XML functions

- Combine XML and relational structures
- Generate structured text, e.g. HTML

```
SELECT XML2CLOB(  
    XMLELEMENT(NAME "HTML",XMLELEMENT(NAME "BODY",  
    XMLELEMENT(NAME "TABLE",  
        XMLATTRIBUTES('0' AS "BORDER"),  
        XMLELEMENT(NAME CAPTION, 'USERNAME REPORT'),  
    XMLAGG(XMLCONCAT(  
        XMLELEMENT(NAME TR,XMLELEMENT(NAME TD, 'USERNAME'),  
        XMLELEMENT(NAME TD, USERNAME)),  
    ...  
    XMLELEMENT(NAME TR,XMLELEMENT(NAME TD, 'MESSAGE'),  
        XMLELEMENT(NAME TD, M.MSG_TEXT)),  
        XMLELEMENT(NAME TR, XMLELEMENT(NAME TD, '*****'))  
    ))))  
FROM REPORT_TB U, MSG_TB M
```



# HTML output

```
<HTML><BODY><TABLE BORDER="0"><CAPTION>USERNAME REPORT</CAPTION>
<TR><TD>USERNAME</TD><TD>PETER    </TD></TR>
<TR><TD>START-TS</TD><TD>2006-08-30T10:22:09.208878</TD></TR>
<TR><TD>LAST-REPORT-TS</TD><TD>2006-08-30T10:22:09.209469</TD></TR>
<TR><TD>COUNTER</TD><TD>4</TD></TR>
<TR><TD>MESSAGE</TD><TD>
<DESCRIPTION>THIS IS THE REPORT DESCRIPTION 1</DESCRIPTION>
</TD></TR>
<TR><TD>*****</TD></TR>
...
..
```





# HTML output

