

SAP S/4HANA 최적화를 위한 ABAP Code Pushdown Technology

SAP Korea
July, 2016



발표 내용

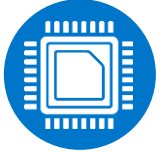
- S/4HANA와 ABAP Code Pushdown Technology
- ABAP Core Data Service
- ABAP Open SQL
- ABAP Managed Database Procedures
- 결언

발표 내용

- **S/4HANA와 ABAP Code Pushdown Technology**
- ABAP Core Data Service
- ABAP Open SQL
- ABAP Managed Database Procedures
- 결언

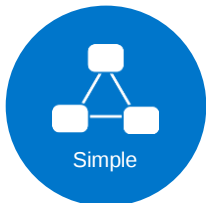
SAP S/4HANA Key Benefit

S/4HANA 장점을 모두 수용할 수 있는 어플리케이션 패러다임 필요



SAP HANA PLATFORM

- IN-MEMORY 컴퓨팅 및 압축
- OLAP & OLTP 통합 처리
- 성능 극대화



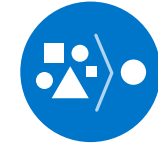
SAP FIORI – SIMPLE USER EXPERIENCE

- SAP FIORI – UX 패러다임
- 간편한 디자인 – 다양한 비즈니스 프로세스간

SAP HANA

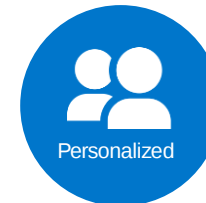
SIMPLIFIED

SAP FIORI



단순화된 어플리케이션 및 데이터 모델

- 요약 및 인덱스 테이블 제거
- 유연성 & 처리량 향상
- 데이터 량 감소



Personalized



Responsive

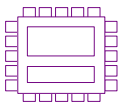


- ROLE-BASED 및 WEB-BASED, 모든 디바이스 지원
- 신속한 응답
- 예측, 시뮬레이션

SAP HANA Platform을 이용한 S/4HANA

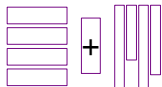
HANA Platform을 최적화를 위한 어플리케이션 변화 요구

SAP HANA Platform 혁신적인 기술 탑재

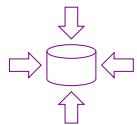


In-Memory Computing

- 실시간 OLTP, OLAP

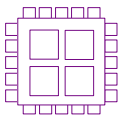


Column 및 Row 스토리지



압축

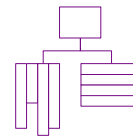
- 5 ~ 50X 압축률 제공



가속

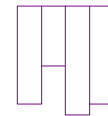
- Parallel 프로세싱으로 10 ~ 10,000 성능 향상

어플리케이션 개발과 수행을 위한 변화 요구



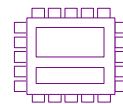
No Aggregate

- On-the-fly 데이터 모델, 데이터 중복 제거



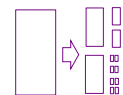
Less Index

- Simple, 그러나 데이터 디렉터리가 인덱스 역할로 우수한 성능 제공



Less 코드 라인

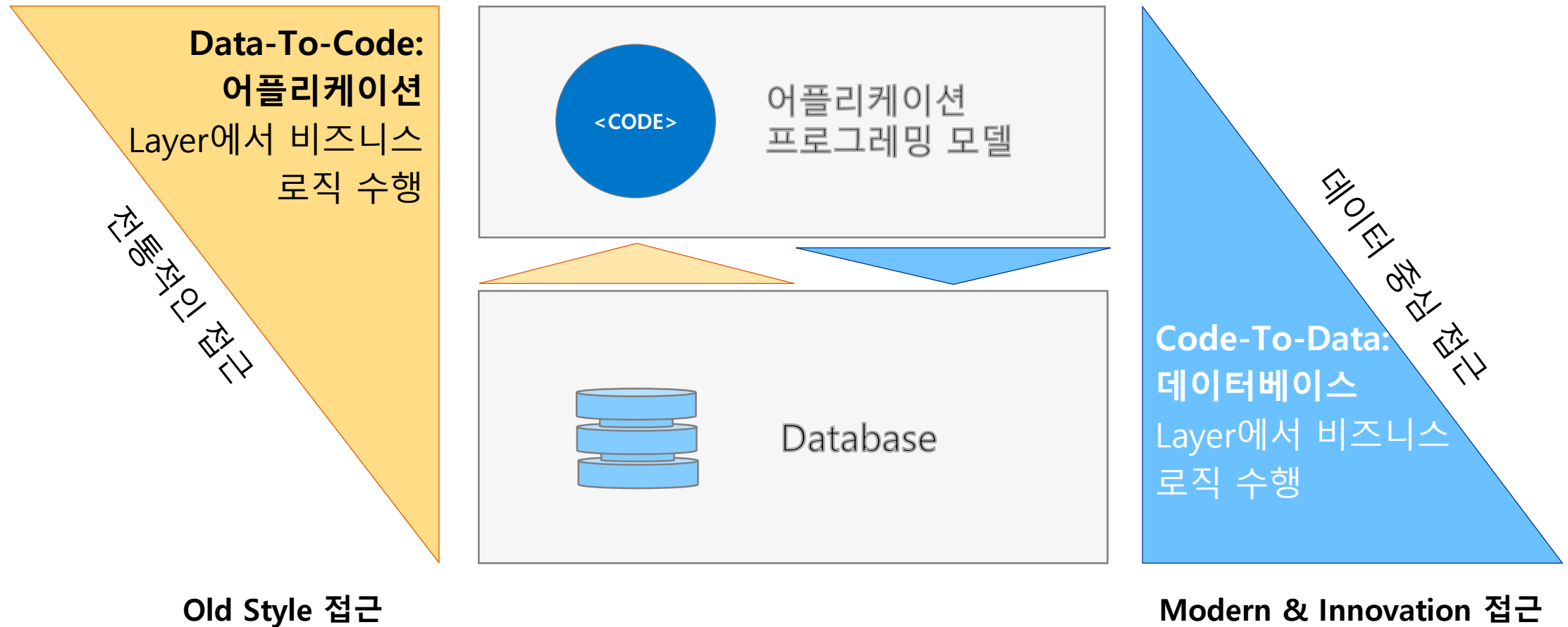
- 데이터 모델과 코드에서 복잡성 제거



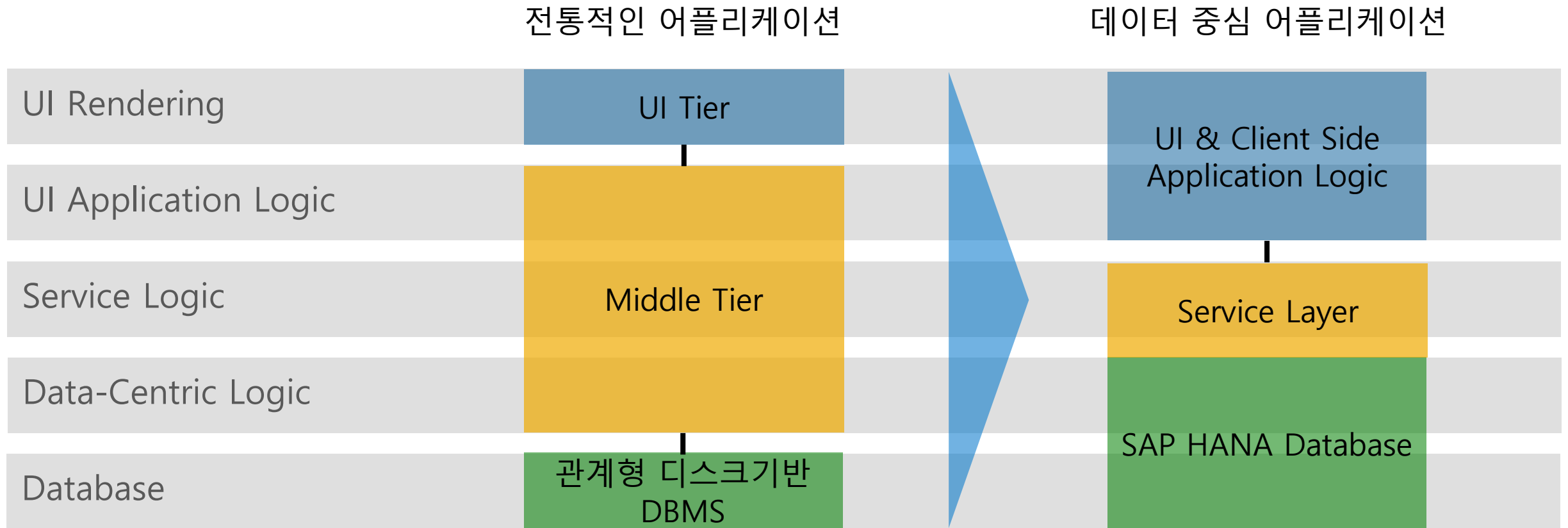
파티셔닝

- 대량 데이터 분석

어플리케이션 패러다임 변화

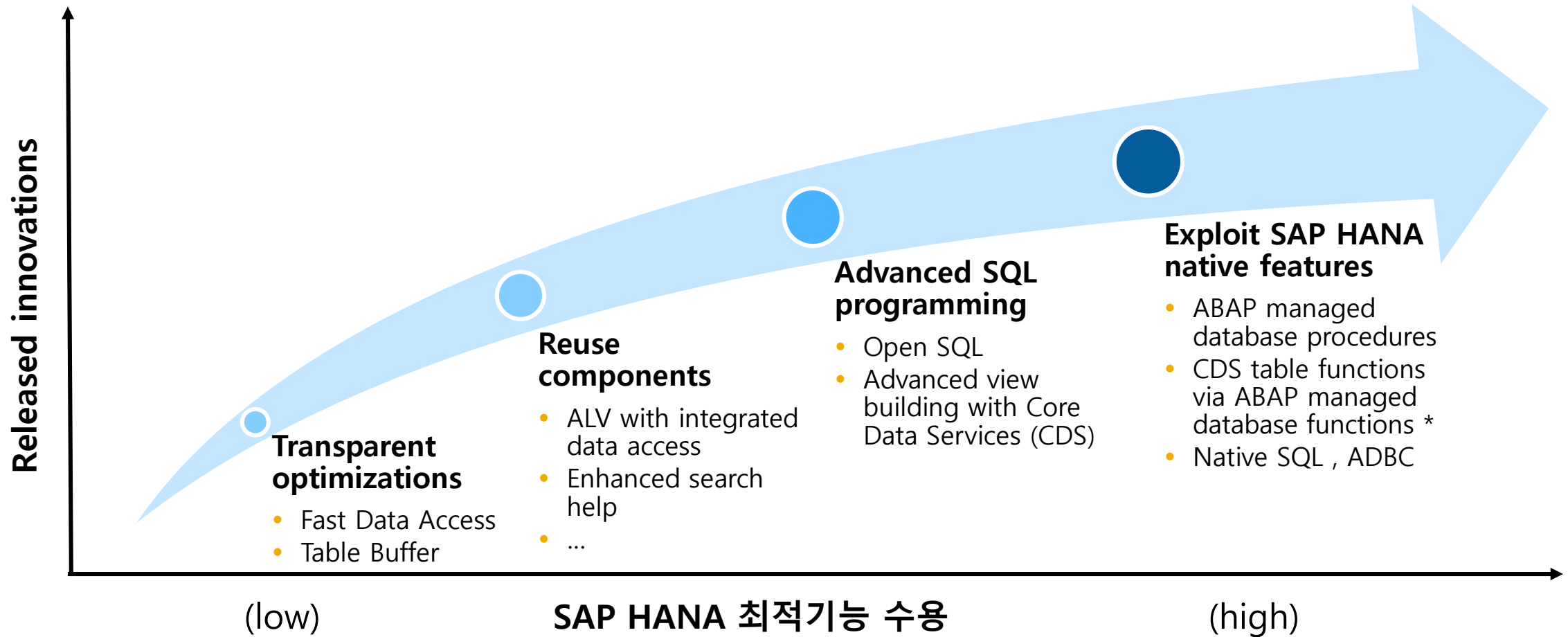


어플리케이션 디자인 변화



데이터베이스 중심 최적화

ABAP Code Pushdown History



* Introduced with ABAP 7.5

ABAP Code Pushdown

ABAP Core Data Services (CDS)

- 신개념 어플리케이션 및 생산성/유연성을 위한 새로운 Data Dictionary View
- S/4HANA Real Time Analytics Architecture

ABAP Open SQL

- 신기능 문법, 표현
- 복잡한 조인, 제한 제거

ABAP Managed Database Procedures (AMDP)

- ABAP Layer에서 HANA SQLScript 수행

CDA Table Functions

- CDS Table Function에서 ADMP 사용



Any Database



HANA
Only

발표 내용

- S/4HANA와 ABAP Code Pushdown Technology
- **ABAP Core Data Service**
- ABAP Open SQL
- ABAP Managed Database Procedures
- 결언

Core Data Services (CDS)

개념

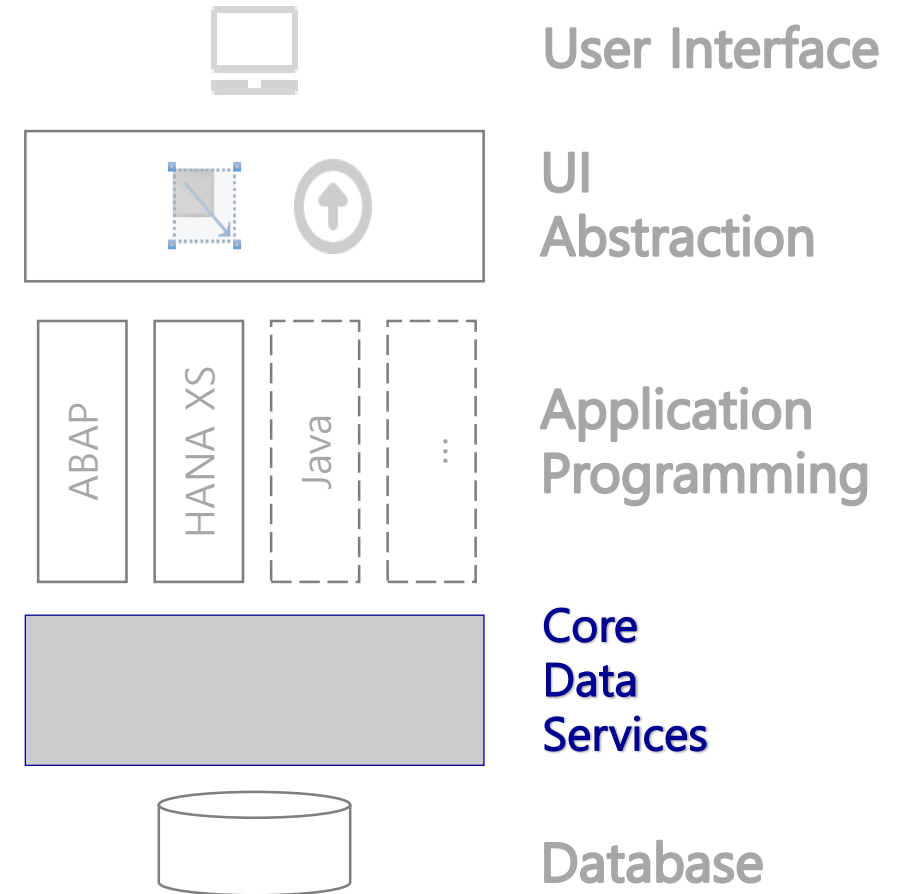
데이터 중심 어플리케이션을 위한 새로운 개념의 데이터 정의와 액세스

CDS 포함

- Data Definition Language (DDL)
- Data Control Language, Data Manipulating Language, ...
- SQL과 비슷하거나 확장
- 서로 다른 프로그래밍 모델간 투명성 제공
- S/4HANA Analytics 핵심

ABAP과 통합

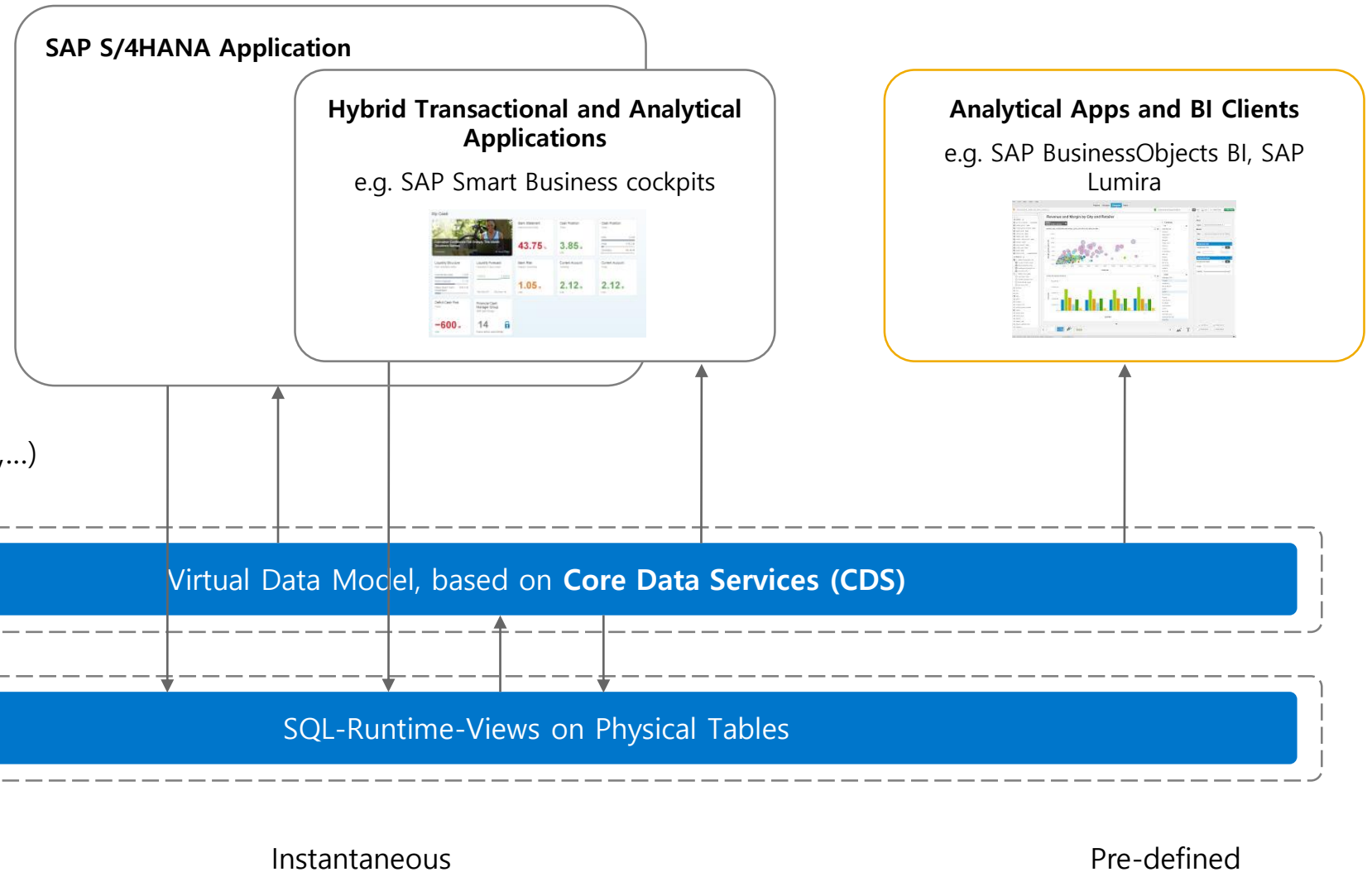
- LM를 통해 간편하게 기존 ABAP / DDIC 인프라와 통합/연동
- CDS Entities를 Open SQL에서 사용

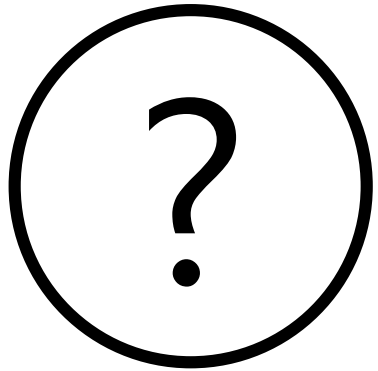


Core Data Services (CDS)

SAP S/4HANA Real Time Analytics Architecture

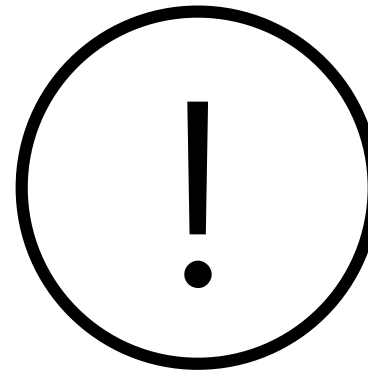
- 지연 X
- 데이터 중복 X
- 즉시 사용 가능한 Content
- SAP BW 시나리오와 통합
- ABAP Lifecycle과 통합
- ERP Authorization 지원
- ERP Hierarchies 지원
- 보다 많은 비즈니스케이스 지원
(analytics, search, fact sheets, planning,...)





Open SQL을 이용하여 모든 업무 수행 가능한가

원칙적으로 가능, 그러나 ...



기능 제한:

- No code Pushdown
- No outer Joins
- No complex Joins
- No unions
- No View on View
-

Simplified, In-Memory Computing, UX 등 지원할 수 있는 어플리케이션 모델 필요

새로운 시나리오

SAP S/4HANA를 위한 ABAP Code Pushdown

ABAP Code Pushdown

확장된 ABAP SQL Interface (DDIC)

OPEN SQL

Core Data Services (CDS)

„Code-Breakouts“
SAP HANA 주요 기능 사용

새로운 ABAP에서 SAP HANA와 인터페이스

ABAP Managed Database Procedure (AMDP)

Table functions (CDS에서 AMDP Call)

ABAP과 같은 개발, 배포, 수정, Troubleshooting

ABAP 개발자의 손쉬운 습득

SAP HANA 자체에서 제공하는 기능 사용

이슈 1: 수행할 일과 SQL 코드간 큰 의미적인 차이

Conceptual Level

Task: 종업원들중 org-unit 4711인 id, name 그리고 그 값에 일치되는 집주소 zip code를 출력

의미적 차이

Implementation Level

SQL: ***SELECT** e.id, e.name, a.zipCode **FROM** Employee e
LEFT OUTER JOIN Employee2Address e2a **ON** e2a.employee = e.id **LEFT OUTER JOIN**
Address a **ON** e2a.address = a.id
AND a.type='homeAddr'
WHERE orgunit=4711;*

이슈 2: SQL 복잡성으로 인한 불합리한 코딩

Conceptual Level

Issue: SQL의 복잡성으로 개발자는 SQL 대신에 Code를 사용

성능 차이: Loop안에서 Loop 사용, 많은 연산이 필요한
중첩 쿼리 사용, ...

Implementation Level

Code:

```
employees = SELECT * FROM Employee WHERE orgunit=4711
foreach c in employees do
    write c.id
    write c.name
    addresses = SELECT * FROM Address a, $Employee2Address c2a
                WHERE a.id = c2a.address AND c2a.employee = :c.id
    foreach a in addresses do
        if a.type = 'homeAddr' then write a.zipCode
    end
end
```

CDS를 이용하여 Conceptual와 Implementation Level로 접근

Conceptual Level

Task: 종업원들중 org-unit 4711인 id, name 그리고 그 값에 일치되는 집주소 zip code를 출력

의미적 차이가 적고,
코딩 최소화

Implementation level

CDS: ***SELECT** id, name, homeAddress.zipCode **FROM** Employee
WHERE orgunit=4711;*

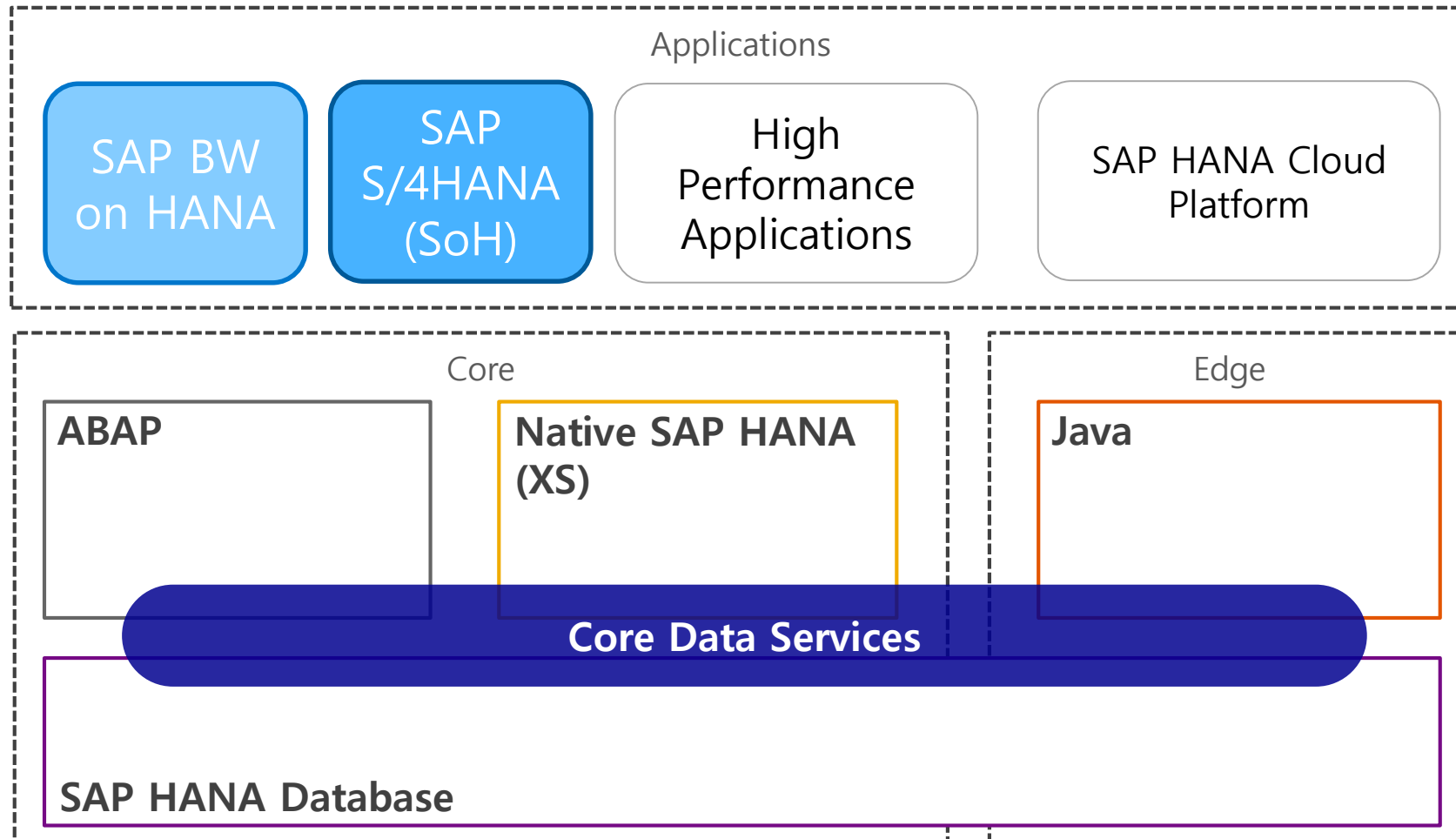
CDS 중요 동기 No. 1

1

CDS는 기술적인 복잡성 대신에 비즈니스 의도를 파악하기 위한 데이터 모델로 SQL 확장

CDS 주요 동기 No. 1

CDS는 지속적으로 SAP 플랫폼을 지원



2

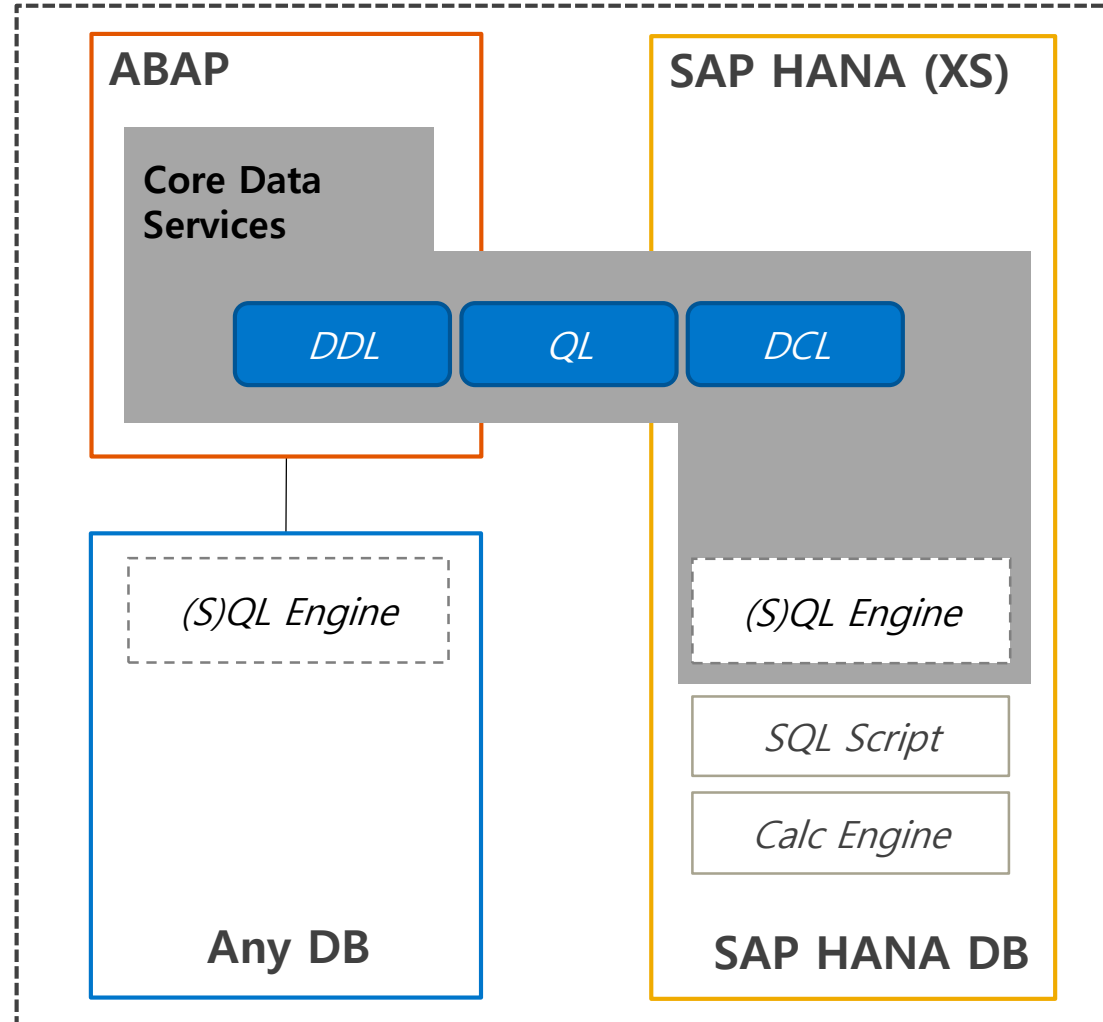
재사용과 호환성을 지원하기 위해 SAP 플랫폼 전반의 프로그래밍 모델과 연동

CDS 주요 동기 No. 2

2가지 Database 통합 방안

ABAP CDS 장점:

- DDIC와 연동
- Lifecycle Management 및 ABAP 인프라와 일괄성 있는 연동
- S/4HANA(Analytics Architecture)에 최적화
- Any DB 지원 (Any DB는 제한적으로 지원, S/4HANA는 HANA기반만 지원)



SAP HANA 장점:

- HANA 코어 엔진과 통합
- In-Memory 기술을 기반으로 성능 최적화
- SAP HANA 기반 제품들과 통합 재사용 (HANA Studio 등)
- No index 등으로 최적화되지 않은 CDS 작성에서도 일정한 성능 제공

CDS 주요 동기 No. 3

3

CDS 최적화된 프로그래밍 모델을 제공

CDS 주요 동기 No. 3

Common Basis for Domain-Specific Frameworks



- 모든 케이스를 위한 단위화된 뷰 모델 및 재사용
- Annotation을 이용하여 다양한 문맥으로 유연성 향상
- ABAP과 SAP HANA간 배포 지원
- 효율적인 개발

CDS 주요 동기 No. 4

4

CDS는 어플리케이션 전반에 통합된 데이터 모델을 제공

CDS 주요 동기 No. 4

SAP HANA 기능, 인프라 활용 및 연동

성능에 민감한 코딩을
어플리케이션 로직에서 구현

```
FOR ALL ENTRIES IN LT_BKPF_0
WHERE BUKRS = LT_BKPF_0-BUKRS
AND BELNR = LT_BKPF_0-BELNR
AND GJAHR = LT_BKPF_0-GJAHR
ORDER BY PRIMARY KEY.
IF SY-SUBRC <> 0.
  RAISE NOT_FOUND.
ENDIF.
ENDIF.

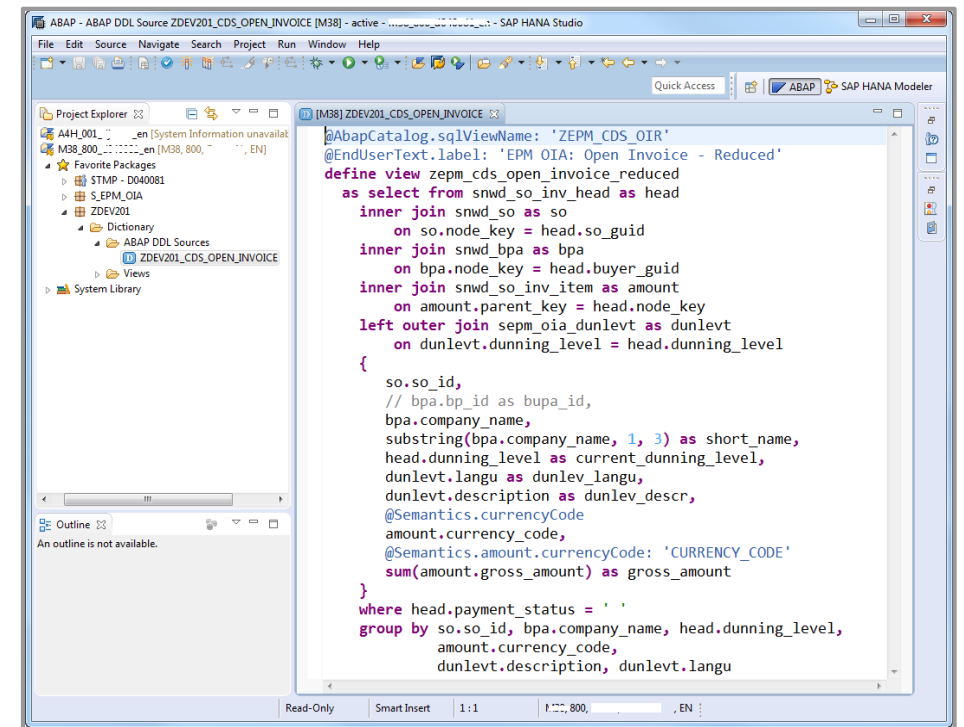
IF NOT LT_BKPF_L[] IS INITIAL.
  SELECT * FROM BSEG_ADD INTO TABLE LT_BSEG_ADD
  FOR ALL ENTRIES IN LT_BKPF_L
  WHERE BUKRS = LT_BKPF_L-BUKRS
  AND BELNR = LT_BKPF_L-BELNR
  AND GJAHR = LT_BKPF_L-GJAHR.
IF SY-SUBRC <> 0.
  RAISE NOT_FOUND.
ENDIF.
ENDIF.

* begin of note 918695
IF NOT LT_BSEG_ADD[] IS INITIAL.
  LOOP AT LT_BSEG_ADD ASSIGNING <LS_BSEG_ADD>.
    CLEAR LS_BSEG.
    MOVE-CORRESPONDING <LS_BSEG_ADD> TO LS_BSEG.
  * convert fkber into bseg format
  IF NOT <LS_BSEG_ADD>-FKBER IS INITIAL.
    CLEAR LS_BSEG-FKBER.
    CALL FUNCTION 'FUNC_AREA_CONVERSION_OUTBOUND'
      EXPORTING
        I_FUNC_AREA_LONG = <LS_BSEG_ADD>-FKBER
      IMPORTING
        E_FUNC_AREA      = LS_BSEG-FKBER
        E_FUNC_AREA_LONG = LS_BSEG-FKBER_LONG.
  ENDIF.
  APPEND LS_BSEG TO LT_BSEG.
ENDIF.
```

ABAP Managed
Code-Pushdown

HANA Studio
ABAP Plug-in
환경에서 개발
및 Activate

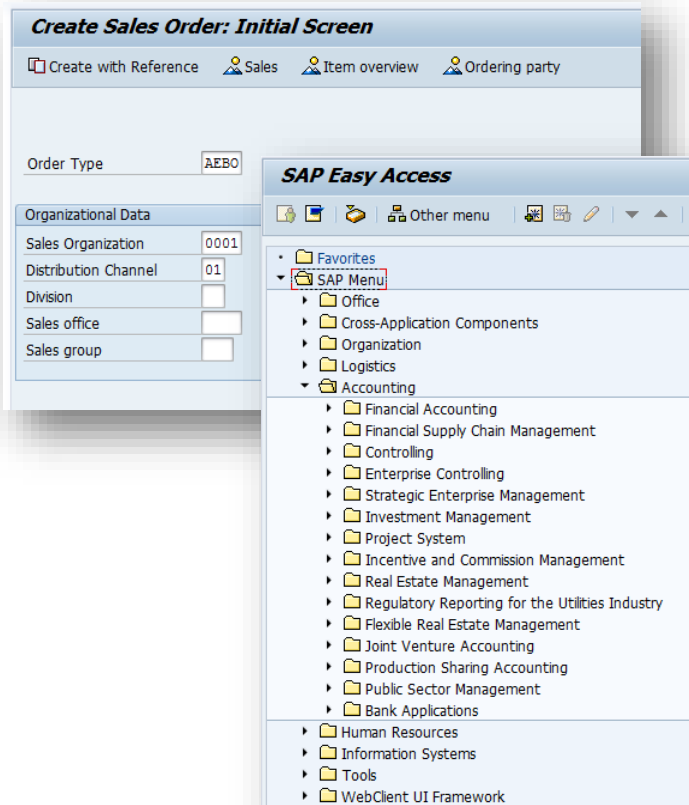
Open SQL, Core Data Services, ABAP
Managed Database Procedures



CDS 주요 동기 No. 4

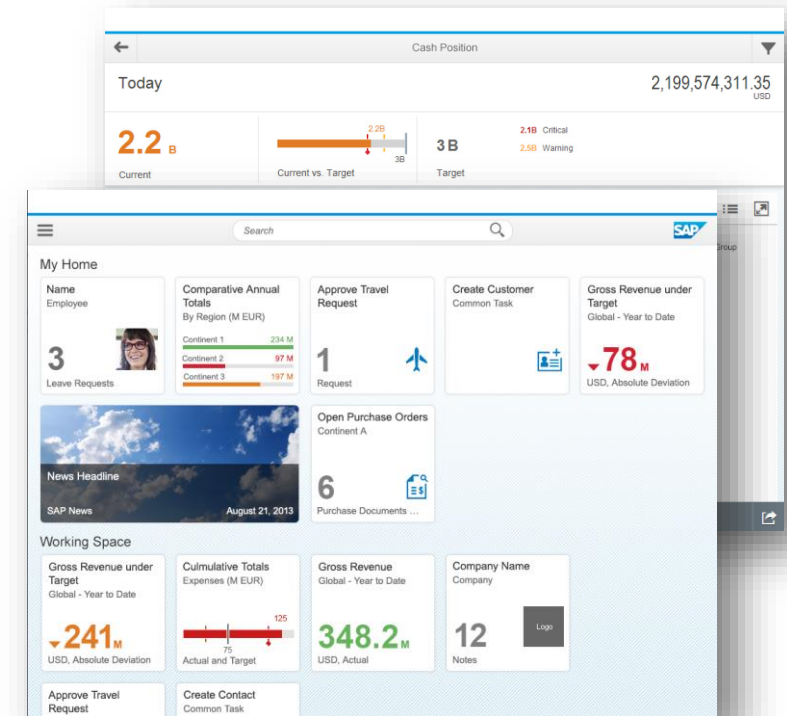
SAP Fiori apps

전통적인 GUI 기반 트랜잭션



SAP Fiori
프로그래밍 모델

Role 기반 SAP Fiori Apps



발표 내용

- S/4HANA와 ABAP Code Pushdown Technology
- ABAP Core Data Service
- **ABAP Open SQL**
- ABAP Managed Database Procedures
- 결언



현대적 문법

&&

표현

&&

완화된 제약

&&

Functions



Open SQL 문법

현대적 문법

```
DATA: lv_user    TYPE c LENGTH 10.
```

```
SELECT node_key,  
       @sy-datum,  
       @lv_user AS username,  
       'X'      AS value
```

```
FROM snwd_so
```

```
INTO @ls_workarea.
```

```
ENDSELECT.
```

Open
SQL

- 호스트 변수 지원 "@"
- 호스트 변수와 리터럴 Select List에서 사용
- Use Cases:
 - DB level Default 변수로 사용 (Code Pushdown)
 - SQL function 파라미터 (Code Pushdown)

Open SQL Expression

- Open SQL 표현들은 DBMS에 의해 검토됨 – Code Pushdown
- SELECT 리스트나 GROUP BY 리스트 또는 SQL(aggregate) Function에서 사용
- 표현 종류

- Elementary expression
- Case expressions
- Arithmetic expressions
- Cast expressions
- String expressions

Case 사용 예)

- ☒ +, -, *, DIV, MOD, ABS
- ☒ +, -, *, ABS, FLOOR, CEIL
- ☒ +, -, *, /, ABS, CAST

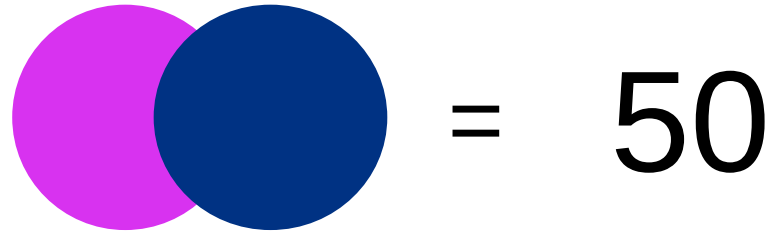
```
SELECT @sy-datum,  
       so_id,  
       CASE lifecycle_status  
         WHEN 'N' THEN 'New'  
         WHEN 'P' THEN  
           CASE billing_status  
             WHEN 'P' THEN 'Paid'  
             ELSE 'Pending'  
           END  
         ELSE 'Undefined'  
       END AS status  
FROM snwd_so  
INTO TABLE @data(itab).
```

Open
SQL

완화된 제약

조인 및 서브쿼리 지원 개수 향상

- 최대 **조인 테이블** 개수 50.



- 최대 **서브쿼리 개수** 9개 -> **50개** 지원



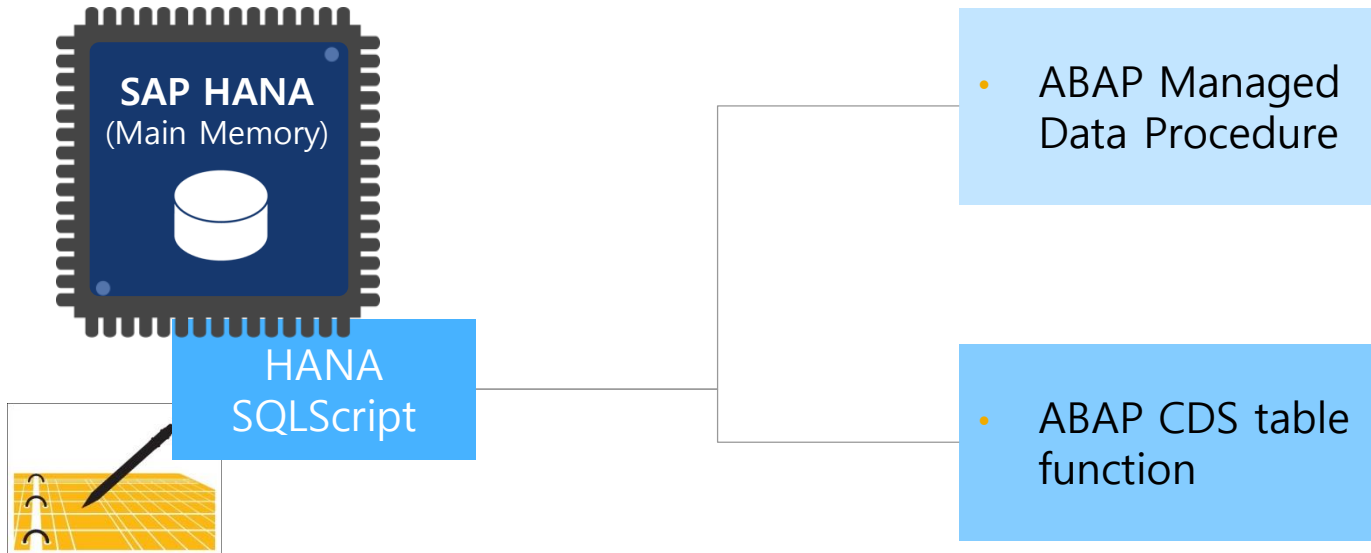
- Open SQL 제한은 ABAP Internal Table 사용의 원인이 됨 – 어플리케이션 복잡 및 성능 저하
- 조인 및 서브쿼리 지원 개수 향상은 Internal Table 사용을 자제하고 내부적으로 Code Pushdown 수행
- S/4 HANA는 이와 같은 인프라를 기반으로 프로그래밍**

발표 내용

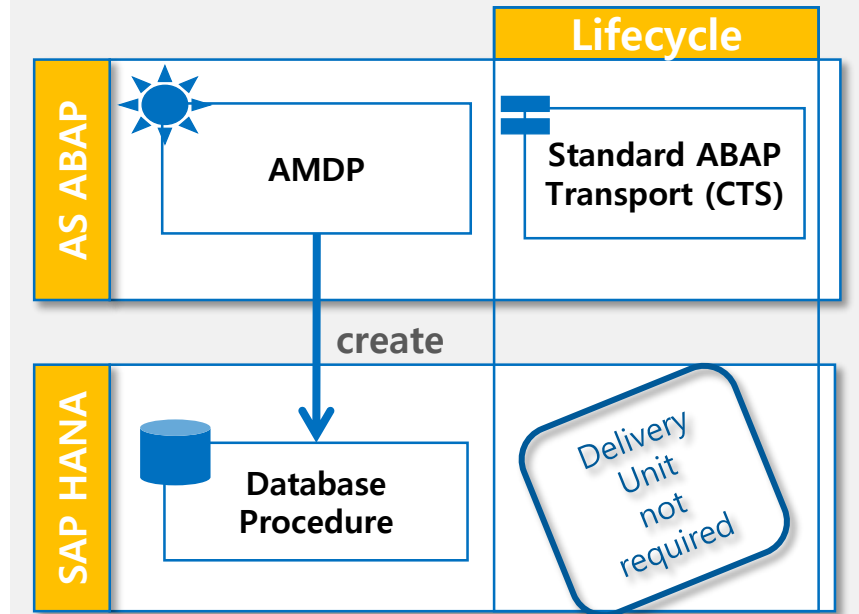
- S/4HANA와 ABAP Code Pushdown Technology
- ABAP Core Data Service
- ABAP Open SQL
- **ABAP Managed Database Procedures**
- 결언

ABAP Managed Database Procedures (AMDP)

개념



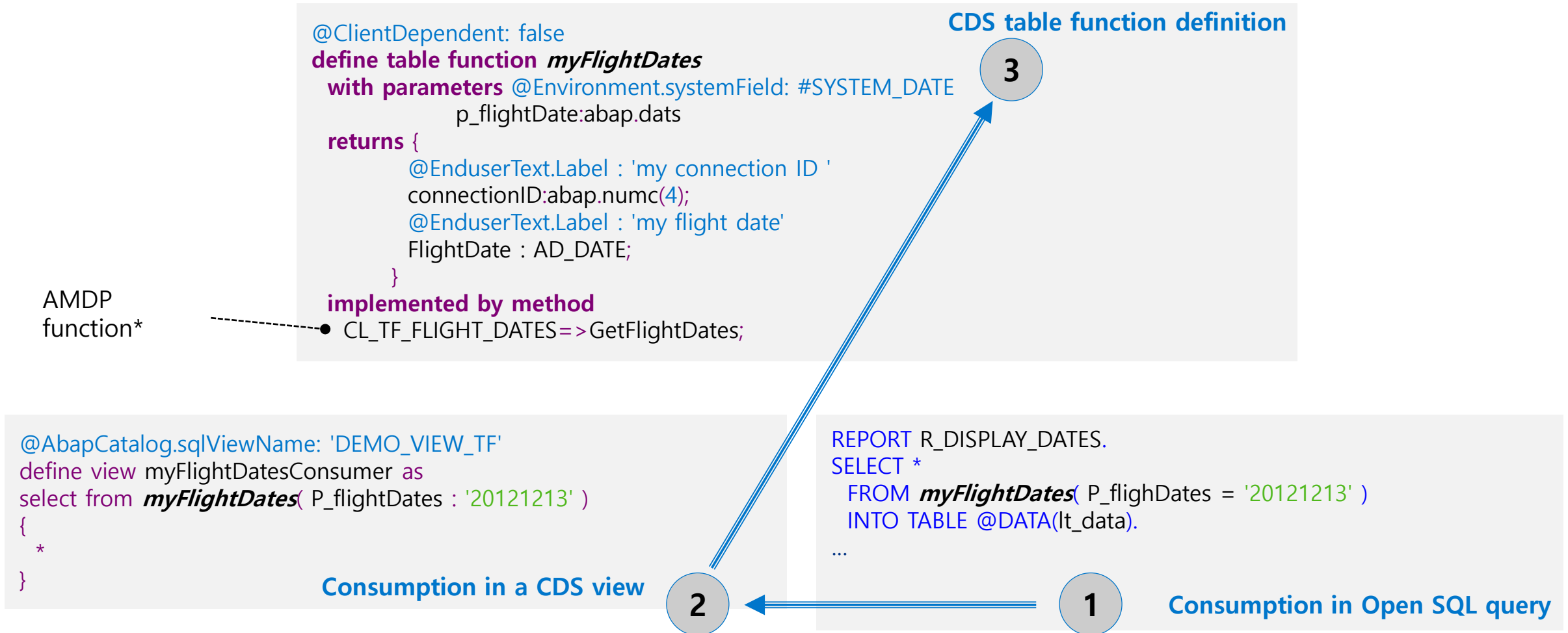
„Top-Down“-Approach



- ABAP Managed Database Procedure는 HANA만 지원
- ABAP에서 HANA SQLScript Stored Procedure를 작성하거나 Call 할 수 있음
- HANA SQLScript을 이용하여 고급 로직을 작성할 수 있으며 HANA에서 로직을 곧바로 처리하여 우수한 성능 제공
- ABAP 개발 툴과 Lifecycle Management를 이용하여 간편한 개발/연동 환경 지원

ABAP Managed Database Procedures (AMDP)

CDS 뷰에서 AMDP 사용 예



Demo

단위 : Microsecond

구분	기존 방식		CDS(Code Pushdown)
	Fast Data Access : Off	Fast Data Access : On	
수행 시간	1,458,944	501,268	38,071
기존 대비 CDS 성능	≒ 38배 향상	≒ 13배 향상	

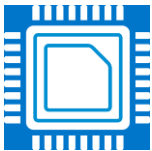
Demo

발표 내용

- S/4HANA와 ABAP Code Pushdown Technology
- ABAP Core Data Service
- ABAP Open SQL
- ABAP Managed Database Procedures
- **결언**

SAP S/4 HANA

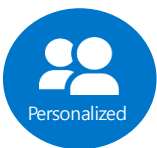
SAP HANA Platform



Simple



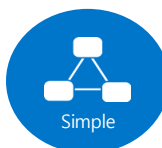
Simple User Experience



Personalized

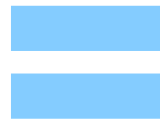


Responsive



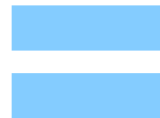
Simple

Open SQL



- 새로운 기능 및 표현
- HANA에 최적화
- S/4HANA에서 Code Pushdown 기술 적용

CDS



- S/4HANA Real-Time Analytics Architecture
- Code Pushdown
- ABAP Lifecycle Management와 연동
- 새로운 개념의 모델 및 어플리케이션

AMDP



- HANA만 지원
- HANA에서 제공하는 고급 및 복잡한 로직 지원
- CDS에서 AMDP 사용
- S/4HANA에서 도입 및 적용 가능

Reference

- ABAP Keyword Documentation : http://help.sap.com/abapdocu_740/en/index.htm?file=abencds_f1_define_view.htm
- ABAP CDS – Core Data Services : <https://wiki.scn.sap.com/wiki/display/ABAP/CDS+-+Core+Data+Services>
- How to Build Core Data Services Views in ABAP on SAP HANA : https://www.youtube.com/watch?v=aYRduGG_kL4
- Code Push Down for HANA Starts with ABAP Open SQL :
<http://scn.sap.com/community/abap/hana/blog/2014/09/26/code-push-down-for-hana-from-abap-starts-with-open-sql>
- ABAP Managed Database Procedure Introduction : <http://scn.sap.com/docs/DOC-51612>
- Implement and consume your first ABAP Managed Database Procedure on HANA : <http://scn.sap.com/docs/DOC-51146>

Rethink the possible

감사합니다.

