GOLDILOCKS 26c Release Note

07228 서울특별시 영등포구 영신로 220 KnK디지털타워 1109호

전화: 02-322-6288 팩스: 02-322-6788

Webpage: www.sunjesoft.com

Support: technet@sunjesoft.com

GOLDILOCKS 26c Release Note

SUNJESOFT Inc.

Copyright 2019-2026 (주)선재소프트
Apache License 버전 2.0의 적용을 받음

차례

1. 개요 1

2. 기능 추가 및 개선 3

2.1 데이터베이스 관리 및 정보 조회 3

병렬 복구 3

병렬 백업 3

병렬 복원 4

DICTIONARY_SCHEMA 역할 정보 View 4

DICTIONARY_SCHEMA 히스토그램 View 4

DICTIONARY_SCHEMA 라이브러리 View 5

DICTIONARY_SCHEMA 트리거 View 5

INFORMATION_SCHEMA 역할 정보 View 6

CHECK_CONSTRAINTS View 6

INFORMATION_SCHEMA 트리거 정보 View 6

성능 및 내부 상태 조회 View 7

RECOVERY_SLAVES 7

히스토그램 생성 제어 프로퍼티 7

BUFFER_LRU_SCAN_PERCENT 8

FULL_TABLE_SCAN_CACHING_THRESHOLD 8

INST_HASH_TABLE_BUCKET_MAX_COUNT 8

GLOBAL_TRANSACTION_LOG_BLOCK_SIZE 9

SHARED_SESSION_MEMORY_INIT_SIZE 9

인스턴트 인덱스 참조 Column 임계값 9

REDO_LOGGING_THROTTLING 9

LOG_FLUSHER_HOT_POLICY_INTERVAL 10

ARCHIVE_LOG_THROTTLING 10

2.2 SQL 및 PSM 11

CURRENT_ROLE 및 LOCAL_MEMBER_POSITION 11

GROUPING 및 GROUPING_ID 11

정규 표현식 함수 및 REGEXP_LIKE 조건 11

JSON 출력의 PRETTY 옵션 12
JSON 객체의 key 중복 제어 12
JSON 배열 집계의 ORDER BY 12
APPROX_COUNT_DISTINCT 12
객체별 페이지 단편화 통계 함수 13
ROLE 객체 DDL 13
ANALYZE TABLE의 Histogram 구축 13
ANALYZE TABLE의 FOR COLUMN GROUPS 14
ANALYZE Sampling의 성능 및 정확도 개선 14
Audit Policy의 Role 감사 14
Change Tracking 파일 이름 변경 14
CHECK 제약 조건 15
FOREIGN KEY 제약 조건 15
제약 조건 활성화 및 비활성화 15
인덱스 활성화 및 비활성화 15
테이블 트리거 실행 순서 지정 16
ALTER TABLE name REORGANIZE 16
MERGE 16
APPEND INSERT 및 힌트 17
DELETE FROM의 local shard limit 17
GROUP BY의 ROLLUP·CUBE·GROUPING SETS 17
GROUP BY에서 SELECT 별칭 참조 17
PIVOT 절 18
UNPIVOT 절 18
TABLESAMPLE 절 18
집계 함수의 FILTER 18
JSON 집계 함수의 FILTER 19
Window 및 UNION ALL 실행 위치 힌트 19
ALTER SYSTEM CANCEL SESSION 19
SET ROLE 19
External Routine 20
Trigger 객체 20
Call Specification 20
Library 객체 DDL 20
Trigger 객체 DDL 21
2.3 API 및 개발 환경 22
JDBC commit_write_mode 연결 속성 추가 22
gpec --parse 옵션 추가 22
Embedded SQL Dynamic SQL Method 4 지원 22

PyDBC Python 3 타입 stub 제공 22
PyDBC Python 3 문자 codec 설정 지원 23
PyDBC Python 3 연결 진단·문자셋 조회·output converter 지원 23
PyDBC connection.maxwrite 지원 23
PyDBC cursor.nextset()을 통한 다중 result set 처리 23
PyDBC procedureColumns() 메타데이터 조회 추가 24
PyDBC fetchval()·cursor iterator·statement cancel 지원 24
2.4 Utility 25
gsql/gsqlnet의 Role DDL 출력 25
gsql/gsqlnet의 Trigger DDL 출력 25
gloader/gloadernet의 APPEND INSERT 옵션 25
2.5 Replication 26
CYCLONE을 이용하여 다른 DBMS로 데이터 이전 26
CYCLONE의 Heartbeat protocol 분리 26
3. 호환성 및 운영 변경 27
3.1 플랫폼·트랜잭션·서버 설정 27
릴리즈 플랫폼 변경 27
클러스터 SERIALIZABLE 지원 중단 27
DEFAULT_INDEX_PCTFREE 기본값 27
DEFAULT_MAXTRANS 기본값 28
BUFFER_DIRTY_PAGE_LIMIT 이름 및 기본값 28
클러스터 통신 버퍼 프로퍼티 변경 28
플랜 캐시 제어 프로퍼티 통합 28
PLAN_CACHE_SIZE 온라인 변경 29
V$SQL_CACHE REF_COUNT 제거 29
AGING_PLAN_INTERVAL 기본값 29
SESSION_MEMORY_INIT_SIZE 기본값 29
클러스터 TRANSACTION_TABLE_SIZE 제약 완화 29
클러스터 데드락 검출 방식 변경 30
클러스터 비동기 replica 적용 설정 변경 30
SHARED_MEMORY_STATIC_SIZE 기본값 30
INST_TABLE_PAGE_SIZE 이름 변경 30
클러스터 commit 통신 분리 31
컨트롤 파일 임시 파일 설정 변경 31
온라인 DDL 프로퍼티 이름 변경 31
3.2 SQL 및 PSM 32
ALTER INDEX COALESCE 동시 실행 제한 32
ADD CONSTRAINT ON SCHEMA 권한 제거 32
신규 사용자의 기본 schema path 변경 32

Online DDL의 shard divisor 이름 변경 33
STORAGE의 MINSIZE 지원 중단 33
인덱스 STORAGE의 MAXSIZE 지원 중단 33
GSI COALESCE 동시 실행 제한 34
Built-in package 설치 방식 변경 34
3.3 API 및 개발 환경 35
ODBC ALTERNATE_LOCATOR 연결 속성 이름 변경 35
ODBC SQL_LONG_VARIABLE_LENGTH_STRUCT 필드 역할 분리 35
JDBC ALTERNATE_LOCATOR 연결 속성 이름 변경 35
Embedded SQL GET CLUSTER_GROUP_ID 구문 이름 변경 35
Embedded SQL LONG VARCHAR 구조체 필드 역할 분리 36
Embedded SQL VARCHAR·VARBINARY 음수 길이 오류 처리 변경 36
PyDBC Python 3 빌드·패키징 변경 36
PyDBC cursor.lastrowid 미지원 값 명시 36
PyDBC Python 3 time zone 타입 반환 방식 변경 37
PyDBC Python 2 time zone 타입 문자열 반환 37
PyDBC TIME·TIMESTAMP 소수초 변환 개선 37
PyDBC decimal precision·scale 처리 및 비유한 값 제한 37
SQLAlchemy 1.4·2.0 지원 38
Hibernate 6·7·8 지원 38
3.4 Utility 39
gloader/gloadernet의 directio-size 옵션 39
gdump의 redo_log 파일 유형 이름 39
gdump의 datafile 파일 유형 이름 39
glocator의 ALTERNATE_LOCATOR 설정 이름 39
glocator 이중화 구성 39
glocator sync 옵션의 인자 제거 40
gagent의 ALTERNATE_LOCATOR 설정 이름 40

1.

개요

이 장에서는 GOLDILOCKS 26c.1에서 추가되거나 변경된 기능을 설명한다. 기능 추가 및 개선 사항은 제품 영역별로 설명하고, 기존 환경에 영향을 미치는 지원 범위·설정·구문 변경 사항은 호환성 및 운영 변경 절에서 정리한다.

각 항목에는 변경 내용을 이해하고 사용하는 데 필요한 User Manual의 설명과 참조 링크를 함께 제공한다.

영역	주요 내용
데이터베이스 관리	병렬 백업·복원·복구, 정보 조회 뷰, 서버 프로퍼티 변경
SQL 및 PSM	Role, 제약 조건, 통계 정보, 집계·JSON·정규 표현식, MERGE, External Routine, Trigger
API	ODBC·JDBC·Embedded SQL 변경, PyDBC 기능 및 데이터 타입 처리, 프레임워크 지원
Utility	gsql DDL 출력, gloader APPEND INSERT, gdump·glocator·gagent 변경
Replication	CYCLONE의 다른 DBMS 연동 및 Heartbeat protocol 분리

2.

기능 추가 및 개선

2.1 데이터베이스 관리 및 정보 조회

병렬 복구

병렬 복구 기능이 추가되었다. ALTER DATABASE RECOVER에서 PARALLEL을 사용하여 복구에 참여하는 slave thread 수를 지정할 수 있으며, 데이터베이스·데이터 파일·테이블스페이스 복구 및 불완전 복구에 사용할 수 있다. 온라인 로그와 archive 로그를 적용하여 데이터 파일을 복구한다.

Slave thread 수는 0~64 로 지정할 수 있다. 0 또는 NOPARALLEL을 지정하면 master thread만 사용한다. 병렬 복구 절 또는 PARALLEL의 수를 생략하면 RECOVERY_SLAVES 값을 따른다. 데이터베이스 전체 완전 복구는 MOUNT 상태와 ARCHIVELOG 모드에서 수행한다.

관련 내용은 **ALTER DATABASE RECOVER**, **recovery slaves clause**를 참조한다.

병렬 백업

병렬 백업 기능이 추가되었다. ALTER DATABASE BACKUP INCREMENTAL의 PARALLEL 절로 백업 thread 수를 지정할 수 있다. 지정 범위는 1~64이며, PARALLEL의 수를 생략하면 1로 설정되고, PARALLEL 절 자체를 생략하면 NOPARALLEL로 동작한다.

실제 병렬 처리 수는 대상 데이터 파일 수와 PIECE 수에 따라 제한된다. 지정한 병렬 수보다 데이터 파일이나 PIECE 수가 적으면 해당 개수만큼 병렬 처리한다.

관련 내용은 **ALTER DATABASE BACKUP**, **parallel clause**를 참조한다.

병렬 복원

병렬 복원 기능이 추가되었다. ALTER DATABASE RESTORE 및 테이블스페이스 복원 구문에서 PARALLEL 을 사용하여 thread 수를 지정하고, 증분 백업의 데이터 파일을 복원할 수 있다. 데이터베이스 전체 복원은 MOUNT 상태에서 수행한다.

병렬 수는 1~64 지정할 수 있으며, PARALLEL의 수를 생략하면 1로 설정된다. PARALLEL 절 자체를 생략하면 NOPARALLEL로 동작한다.

관련 내용은 **ALTER DATABASE RESTORE**, **parallel clause**를 참조한다.

DICTIONARY_SCHEMA 역할 정보 View

역할과 역할에 부여된 권한을 조회할 수 있는 다음 DICTIONARY_SCHEMA view가 추가되었다. 역할 목록, 사용자 및 역할에 부여된 관계, 역할에 부여된 권한, 현재 세션의 활성 역할을 확인할 수 있다.

항목	설명
DBA_ROLES, DBA_ROLE_PRIVS, USER_ROLE_PRIVS	각각 전체 역할, 모든 사용자·역할에 부여된 역할, 현재 사용자에게 부여된 역할을 조회한다.
ROLE_COL_PRIVS, ROLE_DB_PRIVS, ROLE_PACKAGE_PRIVS, ROLE_PROC_PRIVS	역할에 부여된 column·데이터베이스·패키지·루틴 권한을 조회한다.
ROLE_ROLE_PRIVS, ROLE_SCHEMA_PRIVS, ROLE_SEQ_PRIVS	역할 간에 부여된 관계와 스키마·시퀀스 권한을 조회한다.
ROLE_SYS_PRIVS, ROLE_TAB_PRIVS, ROLE_TBS_PRIVS	시스템 권한 (데이터베이스·테이블스페이스·스키마), 테이블 권한, 테이블스페이스 권한을 조회한다.
SESSION_ROLES	현재 세션에서 활성화된 역할을 조회한다.

ROLE_*_PRIVS view는 사용자가 접근할 수 있는 역할에 부여된 권한 정보를 제공한다.

관련 내용은 **DBA_ROLES**, **DBA_ROLE_PRIVS**, **ROLE_COL_PRIVS**, **SESSION_ROLES**를 참조한다.

DICTIONARY_SCHEMA 히스토그램 View

히스토그램 버킷 정보를 조회할 수 있는 다음 DICTIONARY_SCHEMA view가 추가되었다. BALANCE 계열 view는 height-balanced histogram 의 버킷 정보를, FREQUENCY 계열 view는 frequency histogram의 버킷 정보를 조회한다.

항목	설명
ALL_HISTOGRAM_BALANCE,	현재 사용자가 접근할 수 있는 버킷을 조회한다.

항목	설명
ALL_HISTOGRAM_FREQUENCY	
DBA_HISTOGRAM_BALANCE, DBA_HISTOGRAM_FREQUENCY	전체 버킷을 조회한다.
USER_HISTOGRAM_BALANCE, USER_HISTOGRAM_FREQUENCY	현재 사용자가 소유한 버킷을 조회한다.

관련 내용은 **ALL_HISTOGRAM_BALANCE**, **ALL_HISTOGRAM_FREQUENCY**, **DBA_HISTOGRAM_BALANCE**, **USER_HISTOGRAM_BALANCE**를 참조한다.

DICTIONARY_SCHEMA 라이브러리 View

라이브러리 객체와 권한을 조회할 수 있는 다음 DICTIONARY_SCHEMA view가 추가되었다. 라이브러리 객체의 조회 범위와 권한 부여 및 수신 관계를 구분하여 확인할 수 있다.

항목	설명
ALL_LIBRARIES, DBA_LIBRARIES, USER_LIBRARIES	각각 현재 사용자가 접근할 수 있는 라이브러리, 전체 라이브러리, 현재 사용자가 소유한 라이브러리를 조회한다.
ALL_LIBRARY_PRIVS, ALL_LIBRARY_PRIVS_MADE, ALL_LIBRARY_PRIVS_RECD	ALL_LIBRARY_PRIVS는 현재 사용자의 소유·부여·수신 관계와 활성 역할 또는 PUBLIC이 수신자인 권한을 조회한다. ALL_LIBRARY_PRIVS_MADE는 사용자가 소유자 또는 부여자인 권한을, ALL_LIBRARY_PRIVS_RECD는 사용자, 활성 역할 또는 PUBLIC이 수신자인 권한을 조회한다.
DBA_LIBRARY_PRIVS	데이터베이스의 모든 라이브러리 권한 부여 정보를 조회한다.
USER_LIBRARY_PRIVS, USER_LIBRARY_PRIVS_MADE, USER_LIBRARY_PRIVS_RECD	USER_LIBRARY_PRIVS는 현재 사용자가 소유자· 부여자 · 수신자인 권한을 조회한다. USER_LIBRARY_PRIVS_MADE는 현재 사용자가 소유자인 권한을, USER_LIBRARY_PRIVS_RECD는 현재 사용자가 수신자인 권한을 조회한다.
ROLE_LIBRARY_PRIVS	현재 사용자가 접근할 수 있는 역할에 부여된 라이브러리 권한을 조회한다.

관련 내용은 **ALL_LIBRARIES**, **ALL_LIBRARY_PRIVS**, **USER_LIBRARY_PRIVS**, **ROLE_LIBRARY_PRIVS**를 참조한다.

DICTIONARY_SCHEMA 트리거 View

트리거 객체를 조회할 수 있는 DICTIONARY_SCHEMA.ALL_TRIGGERS, DBA_TRIGGERS, USER_TRIGGERS가 추가되었다. ALL_TRIGGERS는 현재 사용자가 접근할 수 있는 테이블의 트리거를, DBA_TRIGGERS는 데이터베이스의 모든 트리거를, USER_TRIGGERS는 현재 사용자가 소유한 트리거를 조회한다.

관련 내용은 **ALL_TRIGGERS**, **DBA_TRIGGERS**, **USER_TRIGGERS**를 참조한다.

INFORMATION_SCHEMA 역할 정보 View

역할과 활성 역할의 권한을 조회할 수 있는 다음 INFORMATION_SCHEMA view가 추가되었다.

항목	설명
ADMINISTRABLE_ROLE_AUTHORIZATIONS	현재 사용자 또는 역할이 WITH ADMIN OPTION을 갖는 역할 부여 정보를 조회한다.
APPLICABLE_ROLES, ENABLED_ROLES	현재 SQL 세션에 적용할 수 있는 역할과 활성화된 역할을 각각 조회한다.
ROLE_COLUMN_GRANTS, ROLE_MODULE_GRANTS, ROLE_ROUTINE_GRANTS, ROLE_TABLE_GRANTS, ROLE_USAGE_GRANTS	현재 활성 역할이 사용할 수 있거나 부여한 column·모듈·루틴·테이블 권한 및 객체 USAGE 권한을 조회한다.

관련 내용은 **ADMINISTRABLE_ROLE_AUTHORIZATIONS**, **APPLICABLE_ROLES**, **ENABLED_ROLES**, **ROLE_COLUMN_GRANTS**를 참조한다.

CHECK_CONSTRAINTS View

CHECK 제약 조건을 조회할 수 있는 INFORMATION_SCHEMA.CHECK_CONSTRAINTS view가 추가되었다. 사용자 또는 역할이 소유한 제약 조건의 소유자·스키마·테이블·이름과 검사 조건을 확인할 수 있다. CHECK_CLAUSE는 CHECK 조건식을 제공하며, NOT NULL 제약 조건인 경우 NULL 을 반환한다.

관련 내용은 **CHECK_CONSTRAINTS**를 참조한다.

INFORMATION_SCHEMA 트리거 정보 View

트리거 정의, 실행 순서, 의존 객체를 조회할 수 있는 다음 INFORMATION_SCHEMA view가 추가되었다.

항목	설명
TRIGGERED_UPDATE_COLUMNS	접근할 수 있는 트리거에서 UPDATE 이벤트 대상으로 지정한 column을 조회한다.
TRIGGERS	사용자 또는 역할이 접근할 수 있는 테이블에 정의된 트리거를 조회한다.
TRIGGER_EVENT_ORDER	동일한 테이블·실행 시점·행/문장 단위·이벤트를 공유하는 트리거의 실행 순서를 조회한다.
TRIGGER_MODULE_USAGE, TRIGGER_ROUTINE_USAGE, TRIGGER_SEQUENCE_USAGE,	트리거가 의존하는 모듈·루틴·시퀀스·테이블을 조회한다.

항목	설명
TRIGGER_TABLE_USAGE	

관련 내용은 **TRIGGERED_UPDATE_COLUMNS**, **TRIGGERS**, **TRIGGER_EVENT_ORDER**, **TRIGGER_TABLE_USAGE**를 참조한다.

성능 및 내부 상태 조회 View

운영 상태를 조회할 수 있는 다음 PERFORMANCE_VIEW_SCHEMA view가 추가되었다.

항목	설명
V$DB_PROPERTY	데이터베이스 생성 시 설정되는 영구 프로퍼티를 조회한다. 데이터베이스 생성 후에는 읽기 전용이다.
V$RELATION	테이블과 인덱스 관계의 유형·사용 가능 여부·할당된 페이지 수 등을 조회한다.
V$TCL_LOGFILE	Transaction commit log 파일의 경로·그룹 상태·크기 등을 조회한다.
V$ALLOCATOR	메모리 할당기의 유형과 설명을 조회한다.
V$CLUSTER_CONNECTION, V$CLUSTER_QUEUE	클러스터 연결과 queue 정보를 조회한다. 클러스터에서만 사용할 수 있다.
V$UNDO_SEGMENT	Undo segment 목록과 할당된 페이지 정보를 조회한다.

관련 내용은 **V$DB_PROPERTY**, **V$RELATION**, **V$TCL_LOGFILE**, **V$ALLOCATOR**, **V$CLUSTER_CONNECTION**, **V$CLUSTER_QUEUE**, **V$UNDO_SEGMENT**를 참조한다.

RECOVERY_SLAVES

병렬 복구에 참여하는 slave thread 수를 제어하는 RECOVERY_SLAVES 프로퍼티가 추가되었다. 기본값은 8이며, 0으로 설정하면 slave thread 없이 master thread만 사용하여 복구한다. ALTER DATABASE RECOVER에서 병렬 복구 절 또는 PARALLEL의 수를 생략하면 RECOVERY_SLAVES 에 설정된 값을 사용한다.

관련 내용은 **RECOVERY_SLAVES**, **<recovery slaves clause>**를 참조한다.

히스토그램 생성 제어 프로퍼티

ANALYZE TABLE 수행 시 히스토그램 버킷 수와 샘플 수를 제어하는 다음 프로퍼티가 추가되었다.

항목	설명
HISTOGRAM_BALANCE_BUCKET_COUNT	Height-balanced histogram의 버킷 수를 지정한다.

항목	설명
HISTOGRAM_BALANCE_MAX_SAMPLE_COUNT	Height-balanced histogram 구축 시 샘플링할 데이터의 최대 개수를 지정한다.
HISTOGRAM_FREQUENCY_BUCKET_COUNT	Frequency histogram 생성 기준이 되는 버킷 수를 지정한다. 필요한 버킷 수가 설정값을 초과하면 생성하지 않는다.

두 BUCKET_COUNT 프로퍼티의 기본값은 0이다. 값이 5 이하이면 해당 히스토그램을 구축하지 않으며, 매뉴얼에서 권장하는 값은 20이다.

관련 내용은 **HISTOGRAM_BALANCE_BUCKET_COUNT**, **HISTOGRAM_BALANCE_MAX_SAMPLE_COUNT**, **HISTOGRAM_FREQUENCY_BUCKET_COUNT**를 참조한다.

BUFFER_LRU_SCAN_PERCENT

디스크 테이블스페이스의 버퍼 관리를 위해 BUFFER_LRU_SCAN_PERCENT 프로퍼티가 추가되었다. 즉시 사용 가능한 free 버퍼가 없을 때 LRU 목록에서 재사용할 버퍼를 찾기 위해 검사할 페이지의 비율을 BUFFER_CACHE_SIZE에 설정된 버퍼 페이지 수 기준으로 지정한다. 기본값은 40% 이다.

관련 내용은 **BUFFER_LRU_SCAN_PERCENT**를 참조한다.

FULL_TABLE_SCAN_CACHING_THRESHOLD

디스크 테이블 전체 스캔 시 버퍼 캐시 사용 여부를 결정하는 FULL_TABLE_SCAN_CACHING_THRESHOLD 프로퍼티가 추가되었다. 테이블이 사용 중인 페이지 수를 BUFFER_CACHE_SIZE 를 기준으로 계산한 임계값과 비교하여 캐싱 여부를 결정한다.

기본값은 20으로 BUFFER_CACHE_SIZE의 2.0% 이하의 페이지를 사용하는 테이블을 캐싱한다. 값이 1000이면 100%를 의미하며, 전체 스캔 대상 테이블을 모두 캐싱한다.

관련 내용은 **FULL_TABLE_SCAN_CACHING_THRESHOLD**를 참조한다.

INST_HASH_TABLE_BUCKET_MAX_COUNT

Hash instant table의 예상 bucket count 상한을 지정하는 INST_HASH_TABLE_BUCKET_MAX_COUNT 프로퍼티가 추가되었다. 세션에서 변경할 수 있으며, ALTER SYSTEM 으로 변경한 값의 적용 방식은 DEFERRED로 정의되어 있다.

관련 내용은 **INST_HASH_TABLE_BUCKET_MAX_COUNT**를 참조한다.

GLOBAL_TRANSACTION_LOG_BLOCK_SIZE

global transaction 로그 파일의 블록 크기를 지정하는 GLOBAL_TRANSACTION_LOG_BLOCK_SIZE 프로퍼티가 추가되었다. 단위는 바이트이며 512, 1024, 2048, 4096 중 하나를 지정한다. 기본값은 512이며 변경할 수 없다.

관련 내용은 **GLOBAL_TRANSACTION_LOG_BLOCK_SIZE**를 참조한다.

SHARED_SESSION_MEMORY_INIT_SIZE

공유 세션의 초기 메모리 크기를 지정하는 SHARED_SESSION_MEMORY_INIT_SIZE 프로퍼티가 추가되었다. 기본값은 786432(768K)이며, dedicated session의 초기 메모리 크기를 설정하는 SESSION_MEMORY_INIT_SIZE와 구분된다.

관련 내용은 **SHARED_SESSION_MEMORY_INIT_SIZE**를 참조한다.

인스턴트 인덱스 참조 Column 임계값

인스턴트 인덱스의 공간 사용량을 줄이는 INSTANT_INDEX_REFERENCE_COLUMN_THRESHOLD 프로퍼티가 추가되었다. 인스턴트 인덱스에 저장하는 column 크기가 설정값 이상이면 해당 column을 참조 방식으로 저장한다.

인스턴트 인덱스 key가 16000 bytes 제한을 초과하는 경우, 이 값을 조정하여 key 크기를 줄일 수 있다. 참조 방식을 사용하면 인덱스 공간을 줄일 수 있지만 성능이 저하될 수 있다.

관련 내용은 **INSTANT_INDEX_REFERENCE_COLUMN_THRESHOLD**를 참조한다.

REDO_LOGGING_THROTTLING

인덱스 구축·재구축 시 로그 기록 속도를 제어하는 REDO_LOGGING_THROTTLING 프로퍼티가 추가되었다. 대량의 로그 기록이 시스템과 온라인 서비스에 미치는 영향을 줄이기 위한 설정이다. 로그 버퍼의 dirty block 수가 설정값을 초과하면 디스크에 flush 될 때까지 기다린다.

관련 내용은 **REDO_LOGGING_THROTTLING**를 참조한다.

LOG_FLUSHER_HOT_POLICY_INTERVAL

gmaster의 log flusher thread가 event를 기다릴 때 수행하는 busy waiting 시간을 조절하는 LOG_FLUSHER_HOT_POLICY_INTERVAL 프로퍼티가 추가되었다. 단위는 마이크로초이며, 기본값은 0으로 busy waiting을 수행하지 않는다. 값을 올리면 로그 버퍼를 디스크에 더 빨리 기록할 수 있지만 CPU 사용량이 증가한다.

관련 내용은 **LOG_FLUSHER_HOT_POLICY_INTERVAL**를 참조한다.

ARCHIVE_LOG_THROTTLING

Redo log archiving의 디스크 I/O를 제어하는 ARCHIVE_LOG_THROTTLING 프로퍼티가 추가되었다. Archive 대상 파일에 복사된 데이터 크기가 설정값을 초과할 때마다 한 번씩 sleep하여 복사 작업의 I/O를 조절한다.

관련 내용은 **ARCHIVE_LOG_THROTTLING**를 참조한다.

2.2 SQL 및 PSM

CURRENT_ROLE 및 LOCAL_MEMBER_POSITION

시스템 정보 함수 CURRENT_ROLE과 LOCAL_MEMBER_POSITION()이 추가되었다. CURRENT_ROLE은 현재 세션에 설정된 role을 반환하며, LOCAL_MEMBER_POSITION()은 사용자 질의를 처리하는 서버의 cluster member position을 반환한다. LOCAL_MEMBER_POSITION()은 cluster system에서 실행 멤버를 확인하는 데 사용할 수 있다.

관련 내용은 **CURRENT_ROLE**, **LOCAL_MEMBER_POSITION**를 참조한다.

GROUPING 및 GROUPING_ID

GROUPING과 GROUPING_ID가 추가되었다. GROUP BY와 함께 사용하며, 각 인자가 현재 grouping key이면 0, 아니면 1로 구성된 비트 값을 정수로 반환한다. 이를 통해 소계·총계 행의 grouping 상태를 구분할 수 있다. GROUPING_ID는 GROUPING과 동일하게 동작하며 인자는 최대 31개까지 지정할 수 있다.

관련 내용은 **GROUPING**, **GROUPING_ID**를 참조한다.

정규 표현식 함수 및 REGEXP_LIKE 조건

정규 표현식을 사용하는 다음 함수와 조건이 추가되었다. 검색 대상 문자열과 pattern을 지정하며, 패턴은 최대 512 byte까지 기술할 수 있다.

이름	용도
REGEXP_COUNT	패턴과 일치하는 횟수를 반환한다.
REGEXP_INSTR	일치 문자열의 시작 위치 또는 그 다음 문자 위치를 반환한다.
REGEXP_LIKE Condition	문자열이 패턴과 일치하는지 검사한다.
REGEXP_REPLACE	일치 문자열을 지정한 문자열로 치환한다.
REGEXP_SUBSTR	패턴과 일치하는 부분 문자열을 반환한다.

관련 내용은 **REGEXP_COUNT**, **REGEXP_INSTR**, **REGEXP_LIKE**, **REGEXP_REPLACE**, **REGEXP_SUBSTR**를 참조한다.

JSON 출력의 PRETTY 옵션

JSON String Constructor의 JSON Output Clause에 PRETTY 옵션이 추가되었다. JSON 문자열의 결과 타입을 지정하는 RETURNING 절 뒤에 PRETTY를 기술하여 출력 형식을 변경한다. 반환 타입은 CHAR(n), VARCHAR(n), LONG VARCHAR 중에서 지정한다.

관련 내용은 **JSON Output Clause**를 참조한다.

JSON 객체의 key 중복 제어

JSON_OBJECT, JSON_OBJECTAGG, JSON_OBJECTAGG() OVER에 JSON key uniqueness constraint 옵션이 추가되었다. WITH UNIQUE KEYS는 하나의 JSON 객체에 중복 key가 있으면 오류를 반환하고, WITHOUT UNIQUE KEYS는 중복을 허용한다. 이를 통해 생성할 JSON 객체의 key 중복 정책을 함수 내에서 지정할 수 있다.

관련 내용은 **JSON Key Uniqueness Constraint**, **JSON_OBJECT**, **JSON_OBJECTAGG**, **JSON_OBJECTAGG() OVER**를 참조한다.

JSON 배열 집계의 ORDER BY

JSON_ARRAYAGG와 JSON_ARRAYAGG() OVER에 JSON array aggregate order by clause 옵션이 추가되었다. 함수 내 ORDER BY 절로 값을 정렬한 후 JSON 배열을 생성한다. ASC·DESC와 NULLS FIRST·NULLS LAST를 지정하며, 기본값은 ASC, NULLS LAST 이다.

관련 내용은 **JSON Array Aggregate Order By Clause**, **JSON_ARRAYAGG**, **JSON_ARRAYAGG() OVER**를 참조한다.

APPROX_COUNT_DISTINCT

근사 집계 함수 APPROX_COUNT_DISTINCT가 추가되었다. NULL이 아닌 값의 중복을 제거한 대략적인 개수를 구하며, HyperLogLog 알고리즘을 사용한다. 대량 데이터에서 고유값 수를 추정할 때 사용하고, 반환값은 COUNT(DISTINCT ...)의 정확한 개수와 다를 수 있다. FILTER를 지정하면 조건을 만족하는 값만 집계한다.

관련 내용은 **APPROX_COUNT_DISTINCT**를 참조한다.

객체별 페이지 단편화 통계 함수

페이지 단편화 정보를 조회하는 TABLE_PHYSICAL_STATS, INDEX_PHYSICAL_STATS, GSI_PHYSICAL_STATS가 추가되었다. TABLE_PHYSICAL_STATS는 일반 테이블을, INDEX_PHYSICAL_STATS는 인덱스를, GSI_PHYSICAL_STATS는 클러스터 테이블의 global secondary index를 분석한다. 대상 이름은 문자열이 아닌 identifier로 지정한다.

세 함수는 VARCHAR로 page, used, fragmented 정보를 반환한다. 선택 인자 sampling_ratio_value는 분석할 페이지 비율을 지정하며 1~100 범위이고 기본값은 100이다. 샘플링을 사용하는 경우 used와 fragmented는 샘플 페이지를 기준으로 계산하고, page는 전체 할당 페이지 수를 반환한다.

관련 내용은 **TABLE_PHYSICAL_STATS**, **INDEX_PHYSICAL_STATS**, **GSI_PHYSICAL_STATS**를 참조한다.

ROLE 객체 DDL

Role을 생성·제거하고 사용자 또는 다른 role에 부여·회수하는 DDL이 추가되었다. 새 role에는 별도의 권한이 자동으로 부여되지 않으므로 필요한 권한을 추가로 부여해야 한다.

구문	기능
CREATE ROLE	Role을 정의한다. CREATE ROLE ON DATABASE 권한이 필요하다.
DROP ROLE	Role을 제거한다.
GRANT role TO	사용자 또는 다른 role에 role을 부여한다.
REVOKE role FROM	부여된 role을 회수한다.

관련 내용은 **CREATE ROLE**, **DROP ROLE**, **GRANT role TO**, **REVOKE role FROM**을 참조한다.

ANALYZE TABLE의 Histogram 구축

ANALYZE TABLE 수행 시 height-balanced histogram과 frequency histogram을 구축할 수 있게 되었다. HISTOGRAM_BALANCE_BUCKET_COUNT를 설정하면 height-balanced histogram을, HISTOGRAM_FREQUENCY_BUCKET_COUNT를 설정하면 frequency histogram을 구축한다. 두 프로퍼티의 기본값은 0이고, 값이 5 이하이면 해당 histogram을 구축하지 않으며 권장값은 20이다. Frequency histogram은 필요한 bucket 수가 설정값을 초과하면 생성하지 않는다.

관련 내용은 **ANALYZE TABLE**, **HISTOGRAM_BALANCE_BUCKET_COUNT**, **HISTOGRAM_FREQUENCY_BUCKET_COUNT**를 참조한다.

ANALYZE TABLE의 FOR COLUMN GROUPS

ANALYZE TABLE에 FOR COLUMN GROUPS 절이 추가되었다. 지정한 column 조합의 NUM_DISTINCT 통계를 구축하여 여러 column을 함께 사용하는 조건의 통계 분석에 활용한다. 2~4개의 column을 지정하며 동일한 column을 중복해서 지정할 수 없다. Column 순서만 다른 조합은 같은 column group으로 취급한다.

관련 내용은 **FOR COLUMN GROUPS**, **ANALYZE TABLE**을 참조한다.

ANALYZE Sampling의 성능 및 정확도 개선

ANALYZE sampling에 repeatable random sampling, approximate NUM_DISTINCT (hyper log 알고리즘), exponent 곡선에 의한 NUM_DISTINCT 보정이 도입되었다. 예를 들어, 10% sampling을 사용하면 TPC-H 22개 질의에서 전수 ANALYZE와 동일한 실행 계획을 얻을 수 있다.

ANALYZE TABLE의 ESTIMATE STATISTICS에서 샘플링할 행 수 또는 비율을 지정할 수 있다. SAMPLE percentage PERCENT의 비율은 1~99의 정수이며, 샘플 행 수가 MIN_SAMPLE_ROW_COUNT보다 작으면 해당 프로퍼티 값을 따른다.

관련 내용은 **ESTIMATE STATISTICS**를 참조한다.

Audit Policy의 Role 감사

CREATE AUDIT POLICY와 ALTER AUDIT POLICY에 role_audit_clause가 추가되었다. ROLES role_name으로 특정 role을 사용하여 실행한 SQL을 감사 대상으로 지정한다. 생성한 정책은 AUDIT POLICY로 활성화해야 하며, 생성·변경에는 AUDIT SYSTEM ON DATABASE 권한이 필요하다. 같은 사용자가 실행한 SQL이라도 객체 소유자 권한으로 실행한 작업과 role 권한으로 실행한 작업의 감사 결과는 다를 수 있다.

관련 내용은 **role_audit_clause**, **ALTER AUDIT POLICY**를 참조한다.

Change Tracking 파일 이름 변경

ALTER DATABASE RENAME CHANGE TRACKING FILE 구문이 추가되었다. Change tracking 파일의 새 이름을 지정할 수 있으며, 데이터베이스가 MOUNT 상태이고 change tracking이 활성화되어 있어야 한다. 실행에는 ALTER DATABASE ON DATABASE 권한이 필요하다. 파일 이름에 절대 경로를 지정하지 않으면 SYSTEM_TABLESPACE_DIR와 결합한 경로를 사용한다.

관련 내용은 **ALTER DATABASE RENAME CHANGE TRACKING FILE**을 참조한다.

CHECK 제약 조건

CHECK 제약 조건이 추가되었다. CREATE TABLE, ALTER TABLE name ADD COLUMN, ALTER TABLE name ADD CONSTRAINT에서 각 row가 만족해야 하는 논리식을 정의한다. 조건 결과가 FALSE이면 위반으로 처리하고, TRUE 또는 UNKNOWN이면 조건을 만족하는 것으로 처리한다.

조건에는 subquery, 다른 객체 참조, 비결정적 표현식, pseudo column을 사용할 수 없으며, 참조 column 수는 최대 32개이다. NULL 허용 여부까지 제한하려면 UNKNOWN 처리 규칙을 고려해야 한다.

관련 내용은 **CHECK 제약 조건**, **CREATE TABLE**, **ALTER TABLE name ADD COLUMN**, **ALTER TABLE name ADD CONSTRAINT**를 참조한다.

FOREIGN KEY 제약 조건

FOREIGN KEY 제약 조건이 추가되었다. CREATE TABLE, ALTER TABLE name ADD COLUMN, ALTER TABLE name ADD CONSTRAINT에서 참조 관계를 정의한다. 외래 키 값은 참조 대상의 PRIMARY KEY 또는 UNIQUE 제약 조건에 존재해야 한다. 단, 외래 키에 NULL이 포함된 경우에는 예외적으로 허용된다.

Column 제약 조건에는 REFERENCES, 테이블 제약 조건에는 FOREIGN KEY (...) REFERENCES 형태를 사용한다. 외래 키를 정의하는 테이블은 base table이어야 하며 temp table과 view에는 정의할 수 없다.

관련 내용은 **FOREIGN KEY 제약 조건**, **CREATE TABLE**, **ALTER TABLE name ADD COLUMN**, **ALTER TABLE name ADD CONSTRAINT**를 참조한다.

제약 조건 활성화 및 비활성화

제약 조건의 활성화 여부를 변경하는 constraint enforcement 옵션을 ALTER TABLE name ALTER CONSTRAINT에서 사용할 수 있게 되었다. ENFORCED는 제약 조건을 활성화하고 NOT ENFORCED는 비활성화한다. 변경할 제약 조건은 제약 조건 이름으로 지정하거나, PRIMARY KEY 또는 column 목록에 대응하는 UNIQUE 제약 조건으로 지정한다.

관련 내용은 **ALTER TABLE name ALTER CONSTRAINT**를 참조한다.

인덱스 활성화 및 비활성화

index enforcement를 제어하는 ALTER INDEX name ENABLE/DISABLE이 추가되었다. ENABLE은 인덱스를 구축하고 활성화하며, DISABLE은 인덱스를 비활성화하고 사용하던 extent를 모두 반납한다. 비활성화된 인덱스는 DML과 SELECT에서 사용하지 않는다.

ENABLE과 ENFORCED, DISABLE과 NOT ENFORCED는 각각 동일한 동작을 수행한다. Key 제약 조건을 위해 생성된 인덱스는 ALTER CONSTRAINT로 관리해야 한다.

관련 내용은 **ALTER INDEX name ENABLE/DISABLE**을 참조한다.

테이블 트리거 실행 순서 지정

ALTER TABLE name SET TRIGGER ORDER가 추가되었다. 테이블에 정의된 트리거 이름을 원하는 순서로 재배열하여 실행 순서를 바꾼다. 일부 트리거만 지정하면 해당 트리거를 지정한 순서대로 먼저 실행하고 나머지 트리거는 기존 순서를 유지한다. 나열한 트리거는 같은 테이블에 속하며 BEFORE/AFTER 시점과 ROW/STATEMENT 실행 단위가 같아야 한다.

관련 내용은 **ALTER TABLE name SET TRIGGER ORDER**를 참조한다.

ALTER TABLE name REORGANIZE

테이블을 물리적으로 재구성하는 ALTER TABLE name REORGANIZE가 추가되었다. ONLINE에서는 INSERT·UPDATE·DELETE를 허용하고 OFFLINE에서는 허용하지 않으며, 기본값은 ONLINE이다. SCAN PARTITION과 PARALLEL로 분할·병렬 처리 옵션을 지정할 수 있다.

Redo log 기록의 기본값은 LOGGING이다. NOLOGGING을 사용하면 재구성 후 서버가 비정상 종료될 때 테이블이 unusable 상태가 될 수 있다. 따라서 NOLOGGING을 사용하는 경우 매뉴얼에 따라 재구성 후 CHECKPOINT를 실행해야 한다.

관련 내용은 **ALTER TABLE name REORGANIZE**를 참조한다.

MERGE

MERGE 구문이 추가되었다. USING으로 지정한 원본과 대상 테이블을 ON 조건으로 비교하여, 일치하면 UPDATE·DELETE·DO NOTHING, 일치하지 않으면 INSERT·DO NOTHING을 수행하도록 기술한다. 여러 변경 작업을 조건별로 한 구문에 표현할 수 있다.

각 대상 행은 WHEN 절을 작성한 순서대로 평가하며, 처음 참이 되는 분기만 실행한다. 별도의 MERGE 권한은 없으며, 참조 column에 대한 SELECT 권한과 구문에 명시한 INSERT·UPDATE·DELETE 작업에 필요한 권한이 필요하다.

관련 내용은 **MERGE**를 참조한다.

APPEND INSERT 및 힌트

APPEND INSERT 방식의 데이터 추가 기능과 이를 지정하는 힌트가 추가되었다. 사용하지 않은 공간이나 새로 할당한 공간에 데이터를 추가하여 대량 적재에 활용한다. 디스크 테이블은 버퍼를 거치지 않고 데이터 파일에 직접 기록한다.

INSERT의 APPEND 힌트, gloader의 관련 옵션 또는 특정 대량 동기화 DDL에서 사용할 수 있다. PARALLEL은 여러 세션의 동시 APPEND INSERT를 허용하고, 이 옵션과 함께 사용하는 NOLOGGING은 기록할 로그를 줄인다. STATEMENT_NOFORCE는 문장 완료 시 사용한 데이터 페이지의 디스크 반영 여부를 지정한다. 지정한 옵션에 따라 참조 무결성 검사나 인덱스 키 반영 여부가 달라질 수 있으므로 적재 방식에 맞는 옵션을 선택해야 한다.

관련 내용은 **APPEND INSERT 방식의 데이터 추가**를 참조한다.

DELETE FROM의 local shard limit

DELETE FROM에 데이터 절체를 위한 local shard limit clause가 추가되었다. Cluster의 sharded table에서 local group에 있는 레코드를 지정한 limit_row_count 만큼 삭제한다. Sharded table 에서만 사용할 수 있으며 limit_row_count는 양수이어야 한다. result offset clause 및 fetch limit clause와 함께 사용할 수 없다.

관련 내용은 **local shard limit clause**를 참조한다.

GROUP BY의 ROLLUP·CUBE·GROUPING SETS

GROUP BY의 grouping element에 ROLLUP, CUBE, GROUPING SETS 지원이 추가되었다. ROLLUP은 나열한 항목을 단계적으로 줄여 소계와 총계를 구한다. CUBE는 모든 grouping 조합을 계산하고, GROUPING SETS는 필요한 조합만 명시한다. 집계 목적에 맞게 소계 구성을 선택할 수 있다.

관련 내용은 **group by clause**를 참조한다.

GROUP BY에서 SELECT 별칭 참조

GROUP BY의 grouping column reference에서 select list alias를 참조할 수 있게 되었다. SELECT 목록의 column에 지정한 별칭을 grouping key로 사용할 수 있다. 동일한 별칭이 여러 개이면 식별할 수 없고, 실제 column 이름과 별칭이 같으면 실제 column 참조가 우선한다.

```
SELECT COUNT(*), t1.c1 AS A1 FROM t1 GROUP BY A1;
```

관련 내용은 **group by clause**를 참조한다.

PIVOT 절

Row 값을 column으로 변환하는 PIVOT 절이 추가되었다. FROM 절의 테이블 참조에 집계 함수, FOR에 pivot column, IN에 값 목록을 지정하여 cross table을 구성한다. 집계 함수는 중첩하여 사용할 수 없고, FOR column의 개수와 IN에 나열한 각 값 조합의 원소 개수가 같아야 한다.

관련 내용은 **pivot clause**를 참조한다.

UNPIVOT 절

Column 값을 row로 변환하는 UNPIVOT 절이 추가되었다. IN에는 변환할 원본 column의 이름을 지정하고, 변환된 값을 저장할 column과 원본 column을 식별할 FOR column 이름을 정의한다. NULL 값을 결과에 포함하려면 INCLUDE NULLS를 지정한다. 이를 생략하면 기본적으로 EXCLUDE NULLS 로 동작한다.

관련 내용은 **unpivot clause**를 참조한다.

TABLESAMPLE 절

테이블에서 일부 row를 무작위로 추출하는 sample 절이 추가되었다. TABLESAMPLE에는 0보다 크고 100 이하인 샘플링 비율을 지정하며, PERCENT ROWS 또는 PERCENT PAGES 를 사용하여 샘플링 단위를 선택한다. 샘플링 단위를 생략하면 기본적으로 PERCENT ROWS가 적용된다. REPEATABLE(seed_value) 를 사용하여 seed 값을 지정할 수 있다. 이를 지정하지 않으면 실행할 때마다 샘플링 결과가 달라질 수 있다.

관련 내용은 **sample clause**를 참조한다.

집계 함수의 FILTER

Aggregate function에 FILTER 기능이 추가되었다. 집계 함수 호출 뒤에 FILTER (WHERE condition)을 지정하면 해당 조건을 만족하는 값만 집계에 포함한다. 하나의 SELECT 문 안에서 각 집계 함수마다 서로 다른 조건을 지정할 수 있다.

```
SELECT COUNT(*), COUNT(*) FILTER (WHERE c1 > 1) FROM t1;
```

관련 내용은 **Aggregate Function**, **COUNT(*) FILTER**를 참조한다.

JSON 집계 함수의 FILTER

JSON aggregate constructor에 FILTER 기능이 추가되었다. JSON_ARRAYAGG 또는 JSON_OBJECTAGG 뒤에 FILTER (WHERE condition)을 지정하면 해당 조건을 만족하는 값만 배열 또는 객체의 집계에 포함된다. 집계 결과는 하나의 JSON 문자열 row 로 반환된다.

관련 내용은 **JSON aggregate Constructor**, **JSON_ARRAYAGG**, **JSON_OBJECTAGG**를 참조한다.

Window 및 UNION ALL 실행 위치 힌트

SQL hint에 window hints와 union all driver hints가 추가되었다. Cluster query에서 window 기능 또는 UNION ALL 구문을 현재 서버에서 처리할지, 각 group의 서버에서 처리할지 지정한다. 해당 driver hint는 standalone 환경에서는 무시된다.

Hint 계열	현재 서버	각 group 서버
window hints	LOCAL_WINDOW	REMOTE_WINDOW
union all driver hints	LOCAL_UNION_ALL	REMOTE_UNION_ALL

관련 내용은 **window hints**, **union all driver hints**를 참조한다.

ALTER SYSTEM CANCEL SESSION

세션에서 실행 중인 작업을 취소하는 ALTER SYSTEM CANCEL SESSION 문이 추가되었다. 취소할 대상의 session ID와 serial number를 지정하며, cluster database에서는 대상 member position도 지정할 수 있다. 이 문을 실행하려면 ALTER SYSTEM ON DATABASE 권한이 필요하다.

관련 내용은 **ALTER SYSTEM CANCEL SESSION**을 참조한다.

SET ROLE

현재 세션의 role을 변경하는 SET ROLE role_name 문이 추가되었다. 현재 사용자에게 적용할 수 있는 role을 선택하거나 ACCESS CONTROL ON DATABASE 권한을 사용하여 role을 설정할 수 있다. SET ROLE NONE을 지정하면 current role 설정이 해제된다. 활성 트랜잭션 중에는 role을 변경할 수 없으며, 변경 후에는 설정된 current role을 기준으로 구문이 실행된다.

관련 내용은 **SET ROLE role_name**을 참조한다.

External Routine

사용자가 작성한 C 프로그램을 실행하는 External Routine 이 추가되었다. Shared library에 저장된 함수를 PSM routine에서 호출하며, external routine process인 gextproc가 library를 로딩·실행하고 결과를 PSM에 반환한다. Library 객체와 call specification을 구성하여 DBMS에서 외부 함수의 기능을 사용한다.

관련 내용은 **External Routine**, **Call Specification**을 참조한다.

Trigger 객체

특정 테이블에서 DML이 수행될 때 자동으로 실행할 동작을 정의하는 trigger 객체가 추가되었다. PSM으로 작성한 동작을 INSERT·UPDATE·DELETE 이벤트와 연결하고, BEFORE/AFTER 시점 또는 ROW/STATEMENT 단위로 실행하도록 정의한다. 활성화된 트리거는 해당 이벤트가 발생하면 자동으로 실행되며, 명시적으로 호출할 수는 없다. 대상 테이블을 삭제하면 종속된 트리거도 함께 삭제된다.

관련 내용은 **Trigger 개요**, **CREATE TRIGGER**를 참조한다.

Call Specification

External C function과 PSM routine 정보를 대응시키는 call specification이 추가되었다. LANGUAGE C, LIBRARY, NAME, PARAMETER 를 사용하여 외부 함수의 이름과 인자·반환값의 데이터 타입을 매핑하고, PSM에서 호출한다. 이를 사용하려면 대상 library에 대한 EXECUTE 권한 또는 매뉴얼에 명시된 동등한 권한이 필요하다.

관련 내용은 **Call Specification**을 참조한다.

Library 객체 DDL

Library 객체를 관리하는 CREATE LIBRARY와 DROP LIBRARY 문이 추가되었다. CREATE LIBRARY는 C 프로그램의 shared library 파일 경로를 스키마 객체 이름에 연결하며, OR REPLACE를 지정하면 기존 객체를 교체할 수 있다. DROP LIBRARY는 library 객체를 제거하고, IF EXISTS를 지정하여 해당 객체가 존재하는 경우에만 삭제하도록 할 수 있다.

관련 내용은 **CREATE LIBRARY**, **DROP LIBRARY**를 참조한다.

Trigger 객체 DDL

Trigger 객체의 생성·삭제·재컴파일·활성화·이름 변경을 위한 다음 DDL이 추가되었다.

구문	기능
CREATE TRIGGER	DML 이벤트와 실행할 동작을 정의한다.
DROP TRIGGER	Trigger를 제거한다.
ALTER TRIGGER name COMPILE	Trigger를 재컴파일한다.
ALTER TRIGGER name ENABLE/DISABLE	Trigger의 활성화 여부를 변경한다.
ALTER TRIGGER name RENAME TO	Trigger 이름을 변경한다.

관련 내용은 **CREATE TRIGGER**, **DROP TRIGGER**, **ALTER TRIGGER name COMPILE**, **ALTER TRIGGER name ENABLE/DISABLE**, **ALTER TRIGGER name RENAME TO**를 참조한다.

2.3 API 및 개발 환경

JDBC commit_write_mode 연결 속성 추가

JDBC에 commit_write_mode 연결 속성이 추가되었다. 이 속성은 트랜잭션 완료 시 생성된 로그를 디스크의 로그 파일에 flush하는 방식을 지정한다. 이 값이 0이면 서버 설정을 따르고, 1이면 wait, 2이면 no wait를 의미한다. 기본값은 0이다.

관련 내용은 **JDBC: 연결 프로퍼티**를 참조한다.

gpec --parse 옵션 추가

Embedded SQL precompiler gpec에 --parse 옵션이 추가되었다. 이 옵션은 입력 소스의 파싱 수준을 지정하며, none은 실행에 필요한 최소한의 구조만 파싱하고, partial은 전처리에 필요한 범위까지 파싱한다. 기본값은 partial 이다. 예를 들어, gpec --parse=none sample.gc 와 같이 사용할 수 있다.

관련 내용은 **Embedded SQL: --parse**를 참조한다.

Embedded SQL Dynamic SQL Method 4 지원

Embedded SQL에서 dynamic SQL method 4를 지원한다. 이 방식은 조회할 열의 개수·타입이나 매개변수 개수를 precompile 시점에 알 수 없는 SQL에 사용한다. ANSI SQL descriptor 방식과 SQLDA 방식을 제공하며, descriptor를 통해 실행 시점의 입력·출력 정보를 지정하여 PREPARE·EXECUTE 또는 커서 작업을 수행한다. 두 방식 모두 struct, SQL_CURSOR 타입과 VARCHAR 타입 배열을 이용한 array operation은 지원하지 않는다.

관련 내용은 **Embedded SQL: Method 4**를 참조한다.

PyDBC Python 3 타입 stub 제공

PyDBC Python 3 패키지에 pygoldilocks.pyi 타입 stub이 추가되었다. IDE와 정적 타입 검사 도구에서 사용할 수 있다.

관련 API는 **PyDBC: API Reference**를 참조한다.

PyDBC Python 3 문자 codec 설정 지원

PyDBC Python 3에서 connection.setencoding()과 setdecoding()을 지원한다. setencoding(encoding, ctype=SQL_CHAR)은 SQL 문과 문자열 매개변수를 바이트로 변환할 때 사용할 codec을, setdecoding(sqltype, encoding, ctype=SQL_CHAR)은 문자 결과를 Python 문자열로 변환할 때 사용할 codec을 설정한다. 해당 설정 인자로는 SQL_CHAR만 사용할 수 있으며, wide character 타입은 지원하지 않는다. 설정은 해당 connection에 유지되며, SQL 실행이나 commit·rollback으로 초기화되지 않는다.

관련 내용은 **PyDBC: 문자 codec 설정**을 참조한다.

PyDBC Python 3 연결 진단·문자셋 조회·output converter 지원

PyDBC Python 3에서 connection.messages, character_set_name() 및 SQL 타입별 output converter를 지원한다. messages에는 연결 작업에서 발생한 DB-API warning이 저장된다. 이 목록은 다음 연결 작업이 시작될 때 초기화되므로, 각 작업 직후 확인해야 한다. character_set_name()은 현재 연결에 설정된 GOLDILOCKS 문자셋 이름을 반환한다.

add_output_converter(sqltype, func)로 변환 함수를 등록하면 해당 연결의 모든 cursor에 적용된다. 변환 함수에는 raw bytes 또는 SQL NULL을 나타내는 None이 전달되고, 함수의 반환값이 조회 결과의 행 값이 된다. 변환 함수의 등록 및 변경 사항은 다음 execute() 호출부터 반영된다. 자세한 내용은 **PyDBC: Output Converter**를 참조한다.

PyDBC connection.maxwrite 지원

대용량 문자·바이너리 매개변수를 처리하기 위한 connection.maxwrite를 지원한다. 이 속성은 매개변수를 직접 바인딩할지 SQLPutData로 전송할지 결정하는 기준 크기를 바이트 단위로 지정한다. 0을 지정하면 드라이버와 데이터 타입에 따른 기본 기준을 사용하며, 사용자 지정 값은 255 이상이어야 한다. 이 속성은 입력 데이터의 전송 방식만 변경하며, 데이터를 자르거나 조회 결과의 크기를 제한하지 않는다.

관련 내용은 **PyDBC: maxwrite**를 참조한다.

PyDBC cursor.nextset()을 통한 다중 result set 처리

Procedure가 반환하는 여러 result set을 cursor.nextset() 을 사용하여 순차적으로 처리할 수 있다. 현재 결과 집합의 행은 fetchone() 등의 fetch 함수로 읽고, nextset()을 호출하면 다음 결과 집합으로 이동한다. 다음 결과 집합이 있으면 True를, 더 이상 없으면 None을 반환한다.

관련 내용은 **PyDBC: nextset()**을 참조한다.

PyDBC procedureColumns() 메타데이터 조회 추가

PyDBC에 cursor.procedureColumns(procedure=None, catalog=None, schema=None)이 추가되었다. 이 method는 ODBC SQLProcedureColumns를 통해 procedure의 IN, OUT, INOUT 매개변수와 반환값·결과 열에 대한 메타데이터를 조회한다. 조회 결과에는 column_name, column_type, data_type, type_name, column_size, ordinal_position 등의 정보가 포함된다.

관련 내용은 **PyDBC: procedureColumns()**를 참조한다.

PyDBC fetchval()·cursor iterator·statement cancel 지원

PyDBC에서 fetchval(), cursor iterator 및 statement cancel을 지원한다. fetchval()은 다음 행의 첫 번째 열 값을 반환하며, 다음 행이 없으면 None을 반환한다. cursor iterator는 결과 행을 하나씩 순차적으로 가져와 처리한다. cursor.cancel()은 해당 cursor에서 실행 중인 statement의 취소를 요청한다. 실제 취소 과정에서 예외가 발생할 수 있으므로 예외 처리를 마친 후 cursor를 재사용할 수 있는지 확인해야 한다.

관련 내용은 **PyDBC: fetchval()**과 **cancel()**을 참조한다.

2.4 Utility

gsql/gsqlnet의 Role DDL 출력

Role 객체의 DDL을 출력하는 ₩ddl_role 명령이 추가되었다. ₩ddl_role name 을 실행하면 해당 role의 생성, 부여 및 주석 관련 구문을 모두 출력한다. 필요한 구문만 출력하려면 CREATE, GRANT, COMMENT 옵션을 지정한다.

관련 내용은 **gsql/gsqlnet의 Role DDL 출력**을 참조한다.

gsql/gsqlnet의 Trigger DDL 출력

Trigger 객체의 DDL을 출력하는 ₩ddl_trigger 명령이 추가되었다. ₩ddl_trigger name 을 실행하면 해당 trigger의 생성 및 주석 관련 구문을 출력한다. CREATE 또는 COMMENT 옵션을 지정하면 해당 구문만 출력할 수 있다. 휴지통에 보관된 객체는 출력 대상에서 제외된다.

관련 내용은 **gsql/gsqlnet의 Trigger DDL 출력**을 참조한다.

gloader/gloadernet의 APPEND INSERT 옵션

APPEND INSERT 방식으로 데이터를 로드하는 기능과 --merge, --skip_index_maintenance, --nologging 옵션이 추가되었다. import 작업에서 --merge EXTENT는 serial APPEND INSERT 방식으로, --merge SEGMENT는 PARALLEL APPEND INSERT 방식으로 데이터를 로드한다.

옵션	사용 방법
--skip_index_maintenance	--merge와 함께 사용한다. 인덱스를 갱신하지 않고 unusable 상태로 설정한다. 생략하면 DEFERRED_INDEX_MAINTENANCE 방식으로 수행한다.
--nologging	--merge SEGMENT와 함께 사용하여 NOLOGGING APPEND INSERT 방식으로 데이터를 로드한다. 생략하면 LOGGING 방식으로 수행한다.

기존 --append 옵션은 로드 전 기존 데이터의 유지·삭제·TRUNCATE 여부를 지정하는 옵션이다. 새로 추가된 --merge 옵션과는 용도가 다르다.

관련 내용은 **gloader/gloadernet의 APPEND INSERT 옵션**을 참조한다.

2.5 Replication

CYCLONE을 이용하여 다른 DBMS로 데이터 이전

CYCLONE 을 통해 Oracle, DB2, MySQL, Tibero로 데이터를 이전하는 기능이 추가되었다. CYCLONE master는 원본 GOLDILOCKS의 redo log 에서 변경 내용을 분석하여 전송하고, slave는 ODBC를 이용하여 대상 DBMS에 변경 내용을 적용한다.

대상 DBMS에 따라 slave의 ORACLE_DRIVER, DB2_DRIVER, MYSQL_DRIVER, TIBERO_DRIVER에 해당 ODBC driver 파일의 경로를 지정한다. MySQL과 DB2의 경우 각각 MYSQL_DATABASE, DB2_DATABASE로 대상 database를 지정한다. SYNC 작업에는 master에 대응하는 해당 SYNC_ORACLE_DRIVER, SYNC_DB2_DRIVER, SYNC_MYSQL_DRIVER, SYNC_TIBERO_DRIVER 설정을 사용한다.

GOLDILOCKS와 대상 DBMS의 column 데이터 타입을 일치시켜야 한다. 다른 DBMS와 연동하는 경우 CYCLONE의 이중화 운영 중 DDL 처리 기능을 사용할 수 없다.

관련 내용은 **CYCLONE의 다른 DBMS로 데이터 이전**을 참조한다.

CYCLONE의 Heartbeat protocol 분리

CYCLONE의 heartbeat protocol이 data protocol과 분리되었다. 연결 유지 시간은 기존 설정인 HEARTBEAT_TIMEOUT 으로 지정한다. Master와 slave에서 사용할 수 있으며, network 단절이나 시스템 장애로 연결이 원활하지 않을 때 연결을 유지하는 최대 시간을 초 단위로 지정한다. 기본값은 30초이며, 최소값은 10초이다.

관련 내용은 **CYCLONE의 Heartbeat protocol 분리**를 참조한다.

3.

호환성 및 운영 변경

3.1 플랫폼·트랜잭션·서버 설정

릴리즈 플랫폼 변경

릴리즈 플랫폼에서 HP, AIX, PPC64가 제거되었다. 현재 매뉴얼의 릴리즈 플랫폼 표에는 서버용 linux-x86_64와 Linux·Windows 계열 클라이언트 플랫폼이 안내되어 있으므로 설치 대상에 맞는 패키지를 확인해야 한다.

관련 내용은 **릴리즈 플랫폼**을 참조한다.

클러스터 SERIALIZABLE 지원 중단

클러스터 시스템에서 트랜잭션 격리 수준 SERIALIZABLE을 더 이상 지원하지 않는다. 클러스터 환경에서 SET TRANSACTION 또는 SET SESSION CHARACTERISTICS AS TRANSACTION으로 격리 수준을 설정할 때는 READ COMMITTED를 사용해야 한다. SET TRANSACTION 으로 설정한 격리 수준은 다음 트랜잭션에 적용되며, 해당 트랜잭션이 종료되면 기본 속성으로 복원된다.

관련 내용은 **SET TRANSACTION의 격리 수준**, **SET SESSION CHARACTERISTICS**의 격리 수준을 참조한다.

DEFAULT_INDEX_PCTFREE 기본값

DEFAULT_INDEX_PCTFREE의 기본값이 10으로 변경되었다. 인덱스 생성 시 PCTFREE를 명시하지 않으면 이 프로퍼티 값이 적용된다. 따라서 기존 생성 스크립트에서 PCTFREE를 생략하는 경우에도 변경된 기본값인 10 이 적용된다.

관련 내용은 **DEFAULT_INDEX_PCTFREE**를 참조한다.

DEFAULT_MAXTRANS 기본값

DEFAULT_MAXTRANS의 기본값이 32로 변경되었다. 테이블이나 인덱스 생성 시 MAXTRANS를 명시하지 않으면 이 프로퍼티 값이 적용된다.

관련 내용은 **DEFAULT_MAXTRANS**를 참조한다.

BUFFER_DIRTY_PAGE_LIMIT 이름 및 기본값

INCREMENTAL_CHECKPOINT_CRITERIA의 이름이 BUFFER_DIRTY_PAGE_LIMIT로 변경되었으며, 기본값은 0으로 변경되었다. 기존 이름은 매뉴얼에서 alias로 안내하고 있다.

이 프로퍼티는 버퍼 캐시에서 갱신된 페이지 수를 기준으로 증분 체크포인트를 수행할 시점을 결정한다. 기본값인 0은 무한대를 의미하므로, 버퍼의 모든 페이지가 갱신되더라도 이 기준에 따른 증분 체크포인트는 수행되지 않는다.

관련 내용은 **BUFFER_DIRTY_PAGE_LIMIT**를 참조한다.

클러스터 통신 버퍼 프로퍼티 변경

CLUSTER_CM_BUFFER_COUNT가 deprecated로 지정되었으며, 디스패처별 통신 버퍼 수를 설정하는 다음 프로퍼티가 추가되었다.

항목	설명
LOCKABLE_DISPATCHER_CM_BUFFER_COUNT	lockable dispatcher의 통신 버퍼 수를 지정한다.
LOCKLESS_DISPATCHER_CM_BUFFER_COUNT	lockless dispatcher의 통신 버퍼 수를 지정한다.
SYNC_DISPATCHER_CM_BUFFER_COUNT	synchronization dispatcher의 통신 버퍼 수를 지정한다.

각 프로퍼티의 기본값은 1이다. ALTER SESSION과 ALTER SYSTEM 을 통한 변경은 허용되지 않는다.

관련 내용은 **LOCKABLE_DISPATCHER_CM_BUFFER_COUNT**, **LOCKLESS_DISPATCHER_CM_BUFFER_COUNT**, **SYNC_DISPATCHER_CM_BUFFER_COUNT**를 참조한다.

플랜 캐시 제어 프로퍼티 통합

MAXIMUM_FLANGE_COUNT가 deprecated 로 지정되어 플랜 캐시는 PLAN_CACHE_SIZE 로만 제어한다. PLAN_CACHE_SIZE 는 플랜 캐시에 사용할 메모리 크기를 바이트 단위로 지정하는 프로퍼티다. 따라서 이 값을 SQL 문의 개수로 해석하지 않도록 주의해야 한다.

관련 내용은 **PLAN_CACHE_SIZE**를 참조한다.

PLAN_CACHE_SIZE 온라인 변경

PLAN_CACHE_SIZE를 서버 재시작 없이 온라인으로 변경할 수 있도록 개선되었다. 이 프로퍼티는 SQL 실행 계획 캐시에 사용할 메모리 크기를 지정한다. 세션 단위 변경은 허용되지 않으며, 시스템 단위 변경만 허용된다.

관련 내용은 **PLAN_CACHE_SIZE**를 참조한다.

V$SQL_CACHE REF_COUNT 제거

V$SQL_CACHE에서 REF_COUNT column이 제거되었다. 이 view는 공유 SQL 실행 계획의 통계를 제공하며 SQL 텍스트, 계획 개수, 계획 크기와 계획 텍스트 등을 조회한다. REF_COUNT를 명시적으로 참조하던 모니터링 query는 변경된 column 목록에 맞게 수정해야 한다.

관련 내용은 **V$SQL_CACHE**를 참조한다.

AGING_PLAN_INTERVAL 기본값

AGING_PLAN_INTERVAL의 기본값이 0에서 3으로 변경되었다. 단위는 초이며, 지정된 시간보다 오래된 SQL plan이 aging 대상이 된다.

관련 내용은 **AGING_PLAN_INTERVAL**을 참조한다.

SESSION_MEMORY_INIT_SIZE 기본값

SESSION_MEMORY_INIT_SIZE의 기본값이 131072(128K)에서 524288(512K)로 변경되었다. 이 프로퍼티는 dedicated session에서 사용할 초기 메모리 크기를 지정한다.

관련 내용은 **SESSION_MEMORY_INIT_SIZE**를 참조한다.

클러스터 TRANSACTION_TABLE_SIZE 제약 완화

클러스터의 모든 멤버가 동일한 TRANSACTION_TABLE_SIZE 값을 가져야 한다는 제약이 제거되었다. 각 멤버의 트랜잭션 테이블 크기를 변경하려면 데이터베이스를 재시작해야 한다.

설정값을 줄이려면 재시작 복구 이후 prepare 된 트랜잭션이 사용한 slot 식별자의 최대값보다 큰 값으로 설정해야 한다. 최대값 이하로 설정할 경우 재시작에 실패할 수 있다.

관련 내용은 **TRANSACTION_TABLE_SIZE**를 참조한다.

클러스터 데드락 검출 방식 변경

클러스터 자체 데드락 검출 기능이 도입됨에 따라 기존 CLUSTER_DEADLOCK_TIMEOUT 프로퍼티는 deprecated 되었다.

클러스터 비동기 replica 적용 설정 변경

클러스터의 replica 반영 방식이 기본적으로 async replica 로 처리되도록 개선되었다. 해당 방식을 사용할 수 없는 환경에서는 시스템 내부에서 자동으로 처리하며, 기존 CLUSTER_ASYNC_REPLICATION 프로퍼티는 deprecate 되었다.

SHARED_MEMORY_STATIC_SIZE 기본값

SHARED_MEMORY_STATIC_SIZE 프로퍼티의 기본값이 800M(838860800바이트)로 변경되었다. 이 프로퍼티는 Shared Static Area(SSA)의 크기를 지정하며, ALTER SESSION과 ALTER SYSTEM 구문으로 변경할 수 없는 속성이다.

관련 내용은 **SHARED_MEMORY_STATIC_SIZE**를 참조한다.

INST_TABLE_PAGE_SIZE 이름 변경

INST_TABLE_BLOCK_SIZE 프로퍼티의 이름이 INST_TABLE_PAGE_SIZE로 변경되었다. 기존 이름은 alias로 계속 지원된다. 이 프로퍼티는 인스턴트 테이블의 페이지 크기를 결정한다. 인스턴트 레코드의 고정 영역이 지정된 페이지 크기를 초과할 경우 최대 레코드 길이 오류가 발생할 수 있다.

관련 내용은 **INST_TABLE_PAGE_SIZE**를 참조한다.

클러스터 commit 통신 분리

클러스터의 commit 관련 프로토콜이 별도의 dispatcher와 queue를 사용하도록 분리 개선되었다. 이에 따라 기존 CLUSTER_COMMIT_STREAM_ISOLATION 프로퍼티는 deprecate 되었다.

운영 상태는 V$CLUSTER_DISPATCHER view에서 dispatcher 유형과 송수신 통계로 조회할 수 있으며, TYPE column에는 commit 항목이 포함된다. 클러스터 queue 목록과 적재 건수는 V$CLUSTER_QUEUE view에서 확인할 수 있다.

관련 내용은 **V$CLUSTER_DISPATCHER**, **V$CLUSTER_QUEUE**를 참조한다.

컨트롤 파일 임시 파일 설정 변경

컨트롤 파일 저장 방식이 임시 파일을 사용하지 않는 구조로 변경됨에 따라 CONTROL_FILE_TEMP_NAME 프로퍼티가 deprecate 되었다. 이 프로퍼티는 기존에 컨트롤 파일 복사본의 임시 저장 디렉터리 및 파일명을 지정하는 데 사용되었다.

관련 내용은 **CONTROL_FILE_TEMP_NAME**을 참조한다.

온라인 DDL 프로퍼티 이름 변경

온라인 DDL 관련 프로퍼티 이름이 다음과 같이 변경되었다. 기존 프로퍼티 이름은 하위 호환성을 위해 alias로 지원된다.

기존 이름	변경된 이름	용도
REBALANCE_BLOCK_READ_COUNT	ONLINE_DDL_BLOCK_READ_COUNT	원격 전송 단위인 block에 포함할 레코드 수를 지정한다.
REBALANCE_SHARD_DIVISOR	ONLINE_DDL_SCAN_PARTITION	Online DDL 실행 시 shard를 분할하여 동기화할 개수를 지정한다.
MAXIMUM_JOURNAL_REPLAY_COUNT	ONLINE_DDL_MAXIMUM_JOURNAL_REPLAY_COUNT	DML 수행으로 발생한 journal log를 반복 반영할 최대 차수를 지정한다.
ONLINE_JOURNAL_REPLAY_THRESHOLD	ONLINE_DDL_JOURNAL_REPLAY_THRESHOLD	남은 journal log 양이 적을 때 EXCLUSIVE lock을 획득하여 마지막 반영 단계로 전환하는 임계값을 지정한다.

관련 내용은 **ONLINE_DDL_BLOCK_READ_COUNT**, **ONLINE_DDL_SCAN_PARTITION**, **ONLINE_DDL_MAXIMUM_JOURNAL_REPLAY_COUNT**, **ONLINE_DDL_JOURNAL_REPLAY_THRESHOLD**를 참조한다.

3.2 SQL 및 PSM

ALTER INDEX COALESCE 동시 실행 제한

ALTER INDEX COALESCE는 다른 ALTER 구문과 동시에 실행할 수 없도록 변경되었다. 22c.1에서 동시에 수행하던 관리 작업이 있다면 실행 순서를 조정해야 한다.

COALESCE는 인접한 인덱스 leaf page를 병합하여 사용 공간을 확보하는 작업이다. DML과 동시 실행은 기존과 동일하게 지원되며, 이번 제한 사항은 다른 ALTER 구문과의 동시 실행에만 적용된다.

관련 내용은 **ALTER INDEX name COALESCE**를 참조한다.

ADD CONSTRAINT ON SCHEMA 권한 제거

ADD CONSTRAINT ON SCHEMA 권한이 제거되었다. 테이블 소유자의 경우 CREATE TABLE 시 자동으로 부여된 ALTER ON TABLE 권한을 사용하여 제약 조건을 추가한다. 비소유자의 경우 대상 테이블에 대한 ALTER ON TABLE 또는 ALTER TABLE ON SCHEMA 권한이 필요하다.

추가된 제약 조건의 소유자는 테이블 소유자가 된다.

관련 내용은 **ALTER TABLE name ADD CONSTRAINT**를 참조한다.

신규 사용자의 기본 schema path 변경

CREATE USER 구문으로 사용자를 생성할 때 기본 적용되는 user schema path에서 PUBLIC 스키마가 제외되도록 변경되었다. 이제 사용자의 전용 스키마만 기본 검색 경로로 지정되므로, 22c.1 이하 버전과 동일한 검색 경로가 필요할 경우 사용자 스키마 뒤에 PUBLIC을 명시적으로 설정해야 한다.

```
ALTER USER u1 SCHEMA PATH (u1, public);
```

스키마 이름을 생략한 객체 참조는 schema path 를 기준으로 해석한다. 단, WITHOUT SCHEMA 옵션으로 생성한 사용자의 기본 경로는 PUBLIC으로 설정된다.

관련 내용은 **CREATE USER의 schema 지정**, **Schema path 변경**을 참조한다.

Online DDL의 shard divisor 이름 변경

Online DDL에서 사용하던 shard divisor 옵션의 이름이 scan partition으로 변경되었다. 기존 DDL 스크립트는 현재 문법인 SCAN PARTITION integer를 사용하도록 수정해야 한다. 이 옵션은 재배치·동기화할 shard의 분할 수를 지정하며, 생략하면 ONLINE_DDL_SCAN_PARTITION 값을 따른다. 적용 대상 DDL은 다음과 같다.

- **ALTER DATABASE MOVE SHARD**
- **ALTER DATABASE REBALANCE**
- **ALTER DATABASE REBALANCE EXCLUDE CLUSTER GROUP**
- **ALTER DATABASE SYNCHRONIZE**
- **ALTER TABLE name MOVE SHARD**
- **ALTER TABLE name REBALANCE**
- **ALTER TABLE name REBALANCE EXCLUDE CLUSTER GROUP cluster_group_list**
- **ALTER TABLE name SYNCHRONIZE**

관련 내용은 **scan partition**을 참조한다.

STORAGE의 MINSIZE 지원 중단

다음 DDL의 STORAGE 절에서 MINSIZE 를 더 이상 지원하지 않는다. 기존 DDL 스크립트를 업그레이드 할 때는 해당 옵션을 제거해야 한다. 테이블의 segment 속성으로는 INITIAL·NEXT·MAXSIZE 가, 인덱스의 segment 속성으로는 INITIAL·NEXT가 있다. 적용 대상 DDL은 다음과 같다.

- **CREATE TABLE**
- **CREATE TABLE AS SELECT**
- **CREATE INDEX**
- **ALTER INDEX name STORAGE**
- **ALTER INDEX name REBUILD**
- **ALTER TABLE name ADD GLOBAL SECONDARY INDEX**
- **ALTER TABLE name ALTER GLOBAL SECONDARY INDEX**
- **ALTER TABLE name ALTER GLOBAL SECONDARY INDEX REBUILD**

관련 내용은 **CREATE TABLE**, **CREATE INDEX**를 참조한다.

인덱스 STORAGE의 MAXSIZE 지원 중단

DDL 문의 STORAGE 절에서 MAXSIZE 사용을 더 이상 지원하지 않는다. 이에 따라 인덱스 및 global secondary index를 생성·변경·재구축하는 기존 스크립트에서 해당 옵션을 제거해야 한다. Segment 속성으로는 INITIAL과 NEXT

가 있다. 적용 대상 DDL은 다음과 같다.

- CREATE INDEX
- ALTER INDEX name STORAGE
- ALTER INDEX name REBUILD
- ALTER TABLE name ADD GLOBAL SECONDARY INDEX
- ALTER TABLE name ALTER GLOBAL SECONDARY INDEX
- ALTER TABLE name ALTER GLOBAL SECONDARY INDEX REBUILD

관련 내용은 **CREATE INDEX, ADD GLOBAL SECONDARY INDEX**를 참조한다.

GSI COALESCE 동시 실행 제한

ALTER TABLE ALTER GLOBAL SECONDARY INDEX COALESCE 구문은 다른 ALTER 구문과 동시에 실행할 수 없도록 변경되었다. 따라서 22c.1에서 동시에 수행하던 관리 작업이 있다면 실행 순서를 조정해야 한다.

이 구문은 지정한 테이블의 global secondary index 에 대해 COALESCE 작업을 수행하여 인덱스 단편화를 제거한다. 작업 수행 시 대상 테이블 이름을 지정해야 하며, 전체 구문은 ALTER TABLE name ALTER GLOBAL SECONDARY INDEX COALESCE이다.

관련 내용은 **ALTER TABLE name ALTER GLOBAL SECONDARY INDEX COALESCE**를 참조한다.

Built-in package 설치 방식 변경

22c.1 이하 버전에서 제공되던 일부 built-in package의 설치 방식이 변경되었다. 필요한 패키지를 사용자가 직접 설치할 수 있도록 SQL 스크립트가 제공된다.
기존 버전에서 사용하던 package를 계속 사용하려면 필요한 패키지의 SQL 스크립트를 설치해야 한다. 각 설치 파일은 $GOLDILOCKS_HOME/admin/packages/ 경로에 있으며, gsql에 SYSDBA 권한으로 접속한 후 --import 옵션을 사용하여 실행한다.

Package	설치 파일
DBMS_LOCK	DBMS_LOCK.sql
DBMS_OUTPUT	DBMS_OUTPUT.sql
DBMS_SQL	DBMS_SQL.sql
DBMS_STANDARD	DBMS_STANDARD.sql

관련 내용은 **DBMS_LOCK 설치**, **DBMS_OUTPUT 설치**, **DBMS_SQL 설치**, **DBMS_STANDARD 설치**를 참조한다.

3.3 API 및 개발 환경

ODBC ALTERNATE_LOCATOR 연결 속성 이름 변경

ODBC 연결 속성 이름이 기존 ALTERNATE_LOCATORS 에서 ALTERNATE_LOCATOR 로 변경되었다. 이 속성은 glocator가 응답하지 않을 때 대체 접속 정보를 제공하는 locator를 지정하는데 사용된다. 기존 설정을 사용하는 경우 odbc.ini 파일의 locator DSN 섹션에서 속성명을 ALTERNATE_LOCATOR=HOST=주소:PORT=포트 형식으로 변경해야 한다.

관련 내용은 **ODBC: locator DSN 키워드**를 참조한다.

ODBC SQL_LONG_VARIABLE_LENGTH_STRUCT 필드 역할 분리

ODBC 비표준 타입에 사용되는 SQL_LONG_VARIABLE_LENGTH_STRUCT의 필드 구성이 변경되었다. buf_len은 arr 버퍼의 용량을, len은 실제 데이터 길이를 나타내며, 단위는 모두 바이트이다.
이 구조체는 SQL_C_LONGVARCHAR 와 SQL_C_LONGVARBINARY 에 사용된다. 응용 프로그램은 버퍼를 할당한 뒤 arr와 buf_len을 설정하고, 입력 데이터를 채운 뒤 실제 길이를 len에 설정하여 바인딩한다.

관련 내용은 **ODBC: SQL_C_LONGVARCHAR 바인딩 예**를 참조한다.

JDBC ALTERNATE_LOCATOR 연결 속성 이름 변경

JDBC 연결 속성 ALTERNATE_LOCATORS 의 이름이 ALTERNATE_LOCATOR 로 변경되었다. JDBC 프로퍼티 설정 시에는 소문자 형태인 alternate_locator를 사용하며, 대체 glocator의 주소를 IP:PORT 형식으로 지정한다. 예를 들어 연결 시 전달할 properties 객체에 prop.setProperty("alternate_locator", "127.0.0.1:42582") 와 같이 설정할 수 있다.

관련 내용은 **JDBC: locator 프로퍼티 설정 예**를 참조한다.

Embedded SQL GET CLUSTER_GROUP_ID 구문 이름 변경

Embedded SQL의 GET GROUPID 구문 이름이 GET CLUSTER_GROUP_ID로 변경되었다. EXEC SQL GET CLUSTER_GROUP_ID INTO :group_id 구문 뒤에 대상 SQL을 지정하여 global connection을 사용하는 클러스터 환경에서 해당 SQL의 group ID를 얻는다. 대상 SQL은 shard key가 설정된 테이블에 대한 DELETE, INSERT, SELECT, UPDATE 문이며, :group_id 변수에는 signed numeric 타입을 사용해야 한다. 대상 SQL은 실제로 실행되지 않고 pre

pare 상태로 캐시되며, 유효하지 않은 ID 값인 -1이 반환될 수 있다.

관련 내용은 **Embedded SQL: EXEC SQL GET CLUSTER_GROUP_ID INTO**를 참조한다.

Embedded SQL LONG VARCHAR 구조체 필드 역할 분리

Embedded SQL에서 LONG VARCHAR를 표현하는 SQL_LONG_VARIABLE_LENGTH_STRUCT의 필드 구성이 변경되었다. buf_len은 arr 버퍼의 용량을, len은 실제 데이터 길이를 나타내며 단위는 모두 바이트이다. 응용 프로그램에서 arr의 메모리를 할당하고 buf_len을 설정한 뒤, 데이터의 실제 길이를 len에 설정해야 한다. 사용이 완료된 메모리는 응용 프로그램에서 직접 해제해야 한다.

관련 내용은 **Embedded SQL: LONG VARCHAR**를 참조한다.

Embedded SQL VARCHAR·VARBINARY 음수 길이 오류 처리 변경

Embedded SQL에서 VARCHAR 또는 VARBINARY 변수를 IN 또는 IN_OUT 매개변수로 사용할 때 len 값이 음수이면 오류가 발생하도록 변경되었다. 따라서 입력 전에 arr에 데이터를 저장하고 실제 데이터 길이를 len에 정확히 설정해야 한다. 매뉴얼 예제에서는 문자열 데이터의 경우 strlen(varstr.arr)를 사용하며, 바이너리 데이터의 경우 복사한 바이트 수를 len에 설정한다.

관련 내용은 **Embedded SQL: VARCHAR 제약**과 **VARBINARY 제약**을 참조한다.

PyDBC Python 3 빌드·패키징 변경

PyDBC Python 3 패키지에 pyproject.toml 기반 빌드 방식과 wheel/sdist 패키징을 적용하였다. PyDBC는 GOLDILOCKS ODBC 라이브러리인 gdlcs를 사용하는 C 확장 모듈이다. Linux 환경에서는 $GOLDILOCKS_HOME/app_dev/pygoldilocks/ver3 디렉터리에서 python3 -m pip install . 명령으로 설치하며, 설치에 사용한 Python과 실행하는 Python의 메이저·마이너 버전이 서로 일치해야 한다.

관련 내용은 **PyDBC: Python 3 설치**를 참조한다.

PyDBC cursor.lastrowid 미지원 값 명시

PyDBC의 cursor.lastrowid는 지원되지 않는 속성으로 항상 None을 반환한다. 따라서 이 속성을 통해서는 삽입된 행의 식별자를 얻을 수 없다.

관련 cursor API는 **PyDBC: Cursor**를 참조한다.

PyDBC Python 3 time zone 타입 반환 방식 변경

PyDBC Python 3에서 TIME WITH TIME ZONE과 TIMESTAMP WITH TIME ZONE 데이터 타입을 UTC offset이 보존되는 timezone-aware 객체로 반환하도록 변경되었다. 각 타입의 반환 형태는 datetime.time과 datetime.datetime이다.

관련 내용은 **PyDBC: Python 3 SQL 값 변환**을 참조한다.

PyDBC Python 2 time zone 타입 문자열 반환

PyDBC Python 2에서는 TIME WITH TIME ZONE과 TIMESTAMP WITH TIME ZONE 데이터 타입을 문자열로 반환한다. 이는 Python 3에서 사용하는 datetime.time 및 datetime.datetime 객체로 반환하는 방식과 다르므로, 두 Python 버전을 모두 지원하는 코드에서는 반환 타입을 구분하여 처리해야 한다.

관련 내용은 **PyDBC: Python 2 SQL 값 변환**을 참조한다.

PyDBC TIME·TIMESTAMP 소수초 변환 개선

PyDBC에서 TIME 및 TIMESTAMP 타입의 fractional second와 microsecond 변환 처리가 개선되었다. 관련 데이터 타입 설명에서 두 SQL 타입은 Python에서 각각 datetime.time과 datetime.datetime 객체로 반환된다. SQL 타입의 소수초 정밀도는 0~6자리이며 기본값은 6자리이다.

관련 내용은 **PyDBC: SQL 값 변환**, **TIME 정밀도**, **TIMESTAMP 정밀도**를 참조한다.

PyDBC decimal precision·scale 처리 및 비유한 값 제한

PyDBC에서 NUMBER, NUMERIC, DECIMAL 타입의 precision 과 scale 처리가 개선되었다. NaN과 Infinity 같은 decimal 값은 허용되지 않는다. PyDBC는 Python 의 decimal 입력을 ODBC SQL_NUMERIC 구조로 전달하며, SQL_NUMERIC 조회값을 decimal.Decimal 객체로 반환한다.

관련 내용은 **PyDBC: 매개변수 전달**과 **SQL 값 변환**을 참조한다.

SQLAlchemy 1.4·2.0 지원

SQLAlchemy 1.4 및 2.0 버전을 지원한다. GOLDILOCKS 용 dialect를 사용하려면 Python 3.8 이상, alembic, py goldilocks 패키지가 필요하다. Dialect 설치 후 create_engine() 함수에 goldilocks://사용자명:비밀번호@호스트:포트 형식의 URL을 전달하여 연결한다.

관련 내용은 **SQLAlchemy: 필수 요건**을 참조한다.

Hibernate 6·7·8 지원

Hibernate 6, 7, 8 버전을 지원한다.

관련 내용은 **Hibernate: GoldilocksDialect 클래스 연동**을 참조한다.

3.4 Utility

gloader/gloadernet의 directio-size 옵션

--directio-size 옵션이 deprecate 되었다. 기존 실행 명령과 스크립트에 이 옵션이 사용되었는지 여부를 확인해야 한다.

gdump의 redo_log 파일 유형 이름

gdump의 파일 유형 이름인 log가 redo_log로 변경되었다. 파일 유형은 덤프할 파일 경로 앞에 지정하며, redo_log는 online 또는 archived redo log file을 텍스트로 출력할 때 사용한다.

관련 내용은 **gdump의 redo_log 파일 유형 이름**을 참조한다.

gdump의 datafile 파일 유형 이름

gdump의 파일 유형 이름인 data가 datafile로 변경되었다. 데이터 파일을 덤프할 때는 gdump datafile 파일경로 형식으로 지정한다. --header는 헤더 정보를, --number와 --fetch는 각각 출력할 page의 시작 번호와 개수를 지정하는 기존 옵션이다.

관련 내용은 **gdump의 datafile 파일 유형 이름**을 참조한다.

glocator의 ALTERNATE_LOCATOR 설정 이름

glocator configuration property의 이름이 ALTERNATE_LOCATORS 에서 ALTERNATE_LOCATOR 로 변경되었다. 이중화 대상이 되는 상대 locator name을 지정하고, 동일한 설정 파일 내의 해당 이름 섹션에 HOST와 PORT를 정의한다.

관련 내용은 **glocator의 ALTERNATE_LOCATOR 설정 이름**을 참조한다.

glocator 이중화 구성

glocator의 다중화 구성이 이중화 구성으로 변경되었다. 두 glocator는 시작 순서에 따라 master와 sub(substitute)로 구분된다. --sync 옵션을 사용하여 나중에 시작하는 sub와 먼저 시작한 master를 연결한다. 동기화 시 두 glocato

r의 데이터를 병합하며, 값이 다른 중복 데이터는 최근에 갱신된 데이터로 덮어쓴다.

관련 내용은 **glocator 이중화 구성**을 참조한다.

glocator sync 옵션의 인자 제거

glocator의 --sync 옵션에 지정하던 인자가 제거되었다. 이제 glocator --start --sync 형식으로 시작하며, configuration file 내에 ALTERNATE_LOCATOR가 미리 설정되어 있어야 한다.

관련 내용은 **glocator sync 옵션의 인자 제거**를 참조한다.

gagent의 ALTERNATE_LOCATOR 설정 이름

gagent configuration property의 이름이 ALTERNATE_LOCATORS 에서 ALTERNATE_LOCATOR 로 변경되었다. [AGENT] 섹션에 alternate locator의 이름을 지정하며, 해당 이름의 섹션에 HOST와 PORT를 설정한다.

관련 내용은 **gagent의 ALTERNATE_LOCATOR 설정 이름**을 참조한다.