
CUBRID 2008 R3.0 Beta

릴리스 노트

목차

1. 개요	5
릴리스 노트 정보	5
릴리스 노트 개정 내역	5
참고 문서	5
버그 리포트 및 사용자 피드백 제공 방법	5
추가 정보 안내	5
2. CUBRID 2008 R3.0 정보	6
CUBRID 2008 R3.0 릴리스 특징	6
지원 플랫폼 및 설치 권장 사양	6
라이선스 안내	7
버전 호환성과 운용성	7
응용 프로그램 및 CUBRID 매니저의 호환성	7
데이터베이스 호환성	7
상호 운용성	7
CUBRID 2008 R3.0의 설치 방법	7
Linux에서 설치	7
Windows에서 설치	7
CUBRID 환경 변수 및 OS 환경 변수 설정	7
CUBRID 2008 R3.0 버전으로 업그레이드하는 방법	8
업그레이드 주의 사항	8
데이터베이스 마이그레이션 절차	9
HA 환경에서 데이터베이스 마이그레이션 절차	10
복제 재구성 방법	11
3. CUBRID 2008 R3.0 버전에서 변경된 사항	12
새로 추가된 기능 - SQL 문법 확장 관련	12
CUBRIDSUS-3590 CREATE TABLE 문 확장 및 테이블 복제 기능 지원	12
CUBRIDSUS-3590 prefix_length에 해당하는 앞 부분 스트링에 대해 인덱스 생성 기능 지원 및 시스템 카탈로그 테이블 변경	12
CUBRIDSUS-3590 INSERT 문법 확장 및 REPLACE 문 지원	13
CUBRIDSUS-3590 SELECT 문, DELETE 문, UPDATE 문에서 대상 레코드의 개수를 제한하는 LIMIT 절 지원	13
CUBRIDSUS-3590 SELECT 문에서 FROM 절 생략 가능	14
CUBRIDSUS-3590 SELECT 문에서 GROUP BY ... HAVING 절의 문법 확장 지원	14
CUBRIDSUS-3590 ALTER TABLE 문법 확장 및 추가할 컬럼 위치 지정 기능 지원	14
CUBRIDSUS-3590 TRUNCATE 문을 사용한 모든 레코드 삭제 기능 지원	15
CUBRIDSUS-3590 외래 키 정의 시 ON UPDATE 및 ON DELETE 연산에 대한 트리거 동작 추가	15
CUBRIDSUS-3590 SQL 레벨에서 PREPARED STATEMENT 실행 기능 지원	15
CUBRIDSUS-3590 SQL 확장 문법 적용을 위한 파라미터 추가 지원	15
새로 추가된 기능 - 연산자와 함수 관련	16
CUBRIDSUS-3591 논리 연산자 확장 지원	16
CUBRIDSUS-3591 비교 연산자 확장 지원	16
CUBRIDSUS-3591 비트 연산자 및 비트 함수 신규 지원	16
CUBRIDSUS-3591 수치 연산 함수의 확장 지원	16
CUBRIDSUS-3591 날짜/시간 함수의 확장 지원 및 출력 포맷의 다양화	18
CUBRIDSUS-3591 문자열 함수의 확장 지원	19
CUBRIDSUS-3591 정보 함수의 확장 지원	20
CUBRIDSUS-3591 조건 연산 함수의 확장 지원	20
새로 추가된 기능 - 기타	21
CUBRIDSUS-3005, 3085 데이터베이스 구동 중에도 공간 정리 작업을 수행할 수 있는 compactdb 유틸리티 옵션 지원	21

CUBRIDSUS-2923, 2961, 2972 PHP API에 20여 개의 함수 추가 지원	21
CUBRIDSUS-2829 insert_execution_mode 파라미터에서 설정할 수 있는 INSERT모드 추가	21
변경 및 개선된 사항	21
CUBRIDSUS-2855 cci_connect() 함수의 동작 방식 변경	22
CUBRIDSUS-3391 TO_CHAR()함수에서 포맷 지정자 'HH', 'HH12'의 범위 변경	22
CUBRIDSUS-3213 두 자리 수 연도 값에 대한 해석 범위 변경	22
CUBRIDSUS-2703 POWER() 함수의 반환 타입을 DOUBLE로 변경	22
CUBRIDSUS-2421 조건절이 항상 FALSE인 특정 질의에서 순차 스캔이 발생되지 않도록 개선	22
CUBRIDSUS-961, 2890, 3030 JDBC 드라이버의 getPrimaryKeys() 및 getColumnDisplaySize() 메소드 지원	22
CUBRIDSUS-2661 JDBC 드라이버에서 Spring 프레임워크의 DataAccessException 지원	23
CUBRIDSUS-1737 Linux 환경에서 CUBRID 실행 스크립트 지원	23
CUBRIDSUS-1562, 1628 일부 테이블만 언로드 시, 지정된 입력 파일에 개행 문자가 없어도 정상 실행되도록 수정	23
CUBRIDSUS-1216 HA 환경에서 마스터 서버의 장애 복구 이후, RW모드의 브로커를 자동 초기화하는 기능 지원	23
CUBRIDSUS-1970 계층 질의문에서 인덱스 스캔을 수행하도록 수정	23
수정된 오류	23
CUBRIDSUS-3194,3399,3434,3435 HA 환경에서 applylogdb 프로세스의 CPU 사용률이 증가하는 오류 수정	23
CUBRIDSUS-3196, 3489 HA 환경에서 applylogdb 프로세스가 반복적으로 재시작되며 복제 진행되지 않는 현상 수정	24
CUBRIDSUS-3106 HA 환경에서 특정 커밋 로그 유실로 인해 복제가 중단되는 오류 수정	24
CUBRIDSUS-2831 HA 환경 구성 직후 클라이언트 모드에서 backupdb 유틸리티 수행 시 작업이 중지되는 오류 수정	24
CUBRIDSUS-2872 HA 환경에서 CUBRID heartbeat 노드 상태 정보 불일치 오류 수정	24
CUBRIDSUS-2120 복제/HA 환경에서 외래 키가 정의된 테이블에 INSERT 수행 시 데이터 불일치 오류 수정	24
CUBRIDSUS-3295, 3191 복제 환경에서 데이터 불일치 오류 수정	24
CUBRIDSUS-3135 복제 환경에서 복제 로그 크기가 비정상적으로 증가하는 오류 수정	24
CUBRIDSUS-2638 특정 상황에서 데이터베이스 복구 작업 수행 후 질의가 정상 수행되지 않는 오류 수정	25
CUBRIDSUS-3083 UNIQUE KEY 컬럼에 중복 값이 삽입되는 오류 수정	25
CUBRIDSUS-2127 조인 조건식 내에서 특정 인덱스 키 값에 대해 OR 연산이 적용되지 않는 오류 수정	25
CUBRIDSUS-2062 외부 조인 수행 시, 질의 결과 오류 수정	25
CUBRIDSUS-423 inline-view 형식의 질의문의 결과에서 컬럼 이름이 출력되지 않는 오류 수정	25
CUBRIDSUS-3337 SELECT 리스트에 ORDER BY 절이 존재하는 상관 질의가 명시되는 경우, 질의 결과 오류 수정	25
CUBRIDSUS-2742 FOR ORDERBY_NUM()가 있는 부 질의가 포함된 경우, 질의 결과 오류 수정	25
CUBRIDSUS-2530 특정 계층 질의문에서 레벨 값 출력 오류 수정	26
CUBRIDSUS-2149 인덱스 컬럼에 대해 ON CACHE OBJECT 속성의 외래 키를 정의하는 경우, 참조 값이 업데이트되지 않는 오류 수정	26
CUBRIDSUS-2603 CREATE SERIAL, ALTER SERIAL 문의 START WITH 옵션 오류 수정	26
CUBRIDSUS-3003 ORDER BY 절을 포함한 부질의를 SEQUENCE로 표현할 때 부질의 결과가 정렬되지 않는 오류 수정	26
CUBRIDSUS-3299 CREATE VIEW 문에서 컬럼 이름을 명시하지 않는 경우, VIEW의 컬럼 타입이 잘못 생성되는 오류 수정	26
CUBRIDSUS-3203, 3224, 3246 CAST 함수를 사용한 명시적 타입 변환 수행 시 오류 수정	26
CUBRIDSUS-3491 멀티쓰레드 환경에서 REUSE_OID 테이블에 INSERT/DELETE 연산을 반복 수행 시, 서버 오류 출력되는 문제 수정	27
CUBRIDSUS-440 트리거가 정의된 클래스를 삭제한 경우, 시스템 카탈로그 뷰에서 트리거 정보가 검색되지 않는 오류 수정	27
CUBRIDSUS-1559 COMMIT 또는 ROLLBACK 이벤트 타입 트리거가 언로드되지 않는 오류 수정	27
CUBRIDSUS-2774 broker_log_top 유틸리티의 동작 오류 수정 및 브로커 로그 포맷 변경	27
CUBRIDSUS-2886 JDBC 를 통한 서버 접속 요청이 집중되는 환경에서 접속 실패하는 현상 수정	27
CUBRIDSUS-3367, 3427 Java Stored Procedure에서 ResultSet 반환 오류 및 cci_cursor_update() 동작 오류 수정	28
CUBRIDSUS-3137 DATETIME 타입 값에 대해 JDBC 드라이버의 getString() 출력 값 오류 수정	28
CUBRIDSUS-3457 컬렉션 타입 컬럼에 대한 cci_col_get() 함수의 동작 오류 수정	28
CUBRIDSUS-3523 CCI 및 PHP 응용 환경에서 cci.log 파일이 생성되는 문제 수정	28
CUBRIDSUS-2225 databases.txt에서 구분자가 tab인 경우에도 cubrid backupdb -sp 옵션이 정상 동작하도록 수정	28
CUBRIDSUS-2221 cub_master 프로세스가 허용하는 연결 시도 요청만 처리하도록 수정	28
CUBRIDSUS-1199 브로커 상태 모니터링 수행 시, 특정 항목 값의 출력 오류 수정	29
CUBRIDSUS-2019 CAS 프로세스의 메모리 크기가 2G 초과 시, 재시작되지 않는 오류 수정	29
CUBRIDSUS-2047 media_failure_support = no로 설정된 경우, 임시 보관 로그 크기가 무한 증가하는 오류 수정	29
CUBRIDSUS-343 CSQL에서 50 라인 이상 질의문에 대한 이전 질의문 불러오기 기능의 오류 수정	29

	4
CUBRIDSUS-600 Windows Vista 환경에서 CSQL 세션 명령어 ;edit 실행 시 발생하는 오류 수정.....	29
CUBRIDSUS-3592, 3359 Windows 환경에서 CURBID 설치 오류 수정.....	29
CUBRIDSUS-471 Windows 환경에서 cubrid service stop/start 명령의 실행 결과가 CUBRID 트레이에 반영되지 않는 오류 수정.....	29
4. CUBRID 매니저 2008 R3.0 버전에서 변경된 사항.....	30
새로 추가된 기능	30
HA 기능 확대 지원.....	30
추가된 구문을 위한 신규 키워드 및 UI 메뉴 지원.....	30
[상태 모니터 추가]에서 사용자 설정 기능 지원.....	30
변경 및 개선된 사항.....	30
[새 질의 편집기] 실행 시 원격의 브로커 호스트 정보를 설정할 수 있는 기능 지원.....	30
[질의 자동화]에 사용자 로그인 기능 지원.....	31
cm.conf의 파라미터 값 설정 방식 확장.....	31
특정 경로에 데이터베이스를 복구할 수 있는 기능 지원.....	31
수정된 오류	31
[데이터베이스 언로드] 기능 수행 시, 테이블 일부 언로드가 수행되지 않는 오류 수정.....	31
다중 질의 실행 시 일부 질의 오류에 대한 경고 창이 계속 출력되는 오류 수정.....	31
질의 편집기에서 조회 결과가 5000건을 초과하는 경우 조회를 중단할 수 없는 오류 수정.....	31
특정 상황에서 매니저가 종료되지 않는 오류 수정.....	31
JDK 1.5 버전 환경에서 질의 편집기에서 수행된 질의 결과가 비정상 출력되는 오류 수정.....	32
5. 주의 사항.....	33
CUBRIDSUS-3217 JDBC에서 연결 정보를 URL 스트링으로 입력하는 경우 물음표를 반드시 명시.....	33
CUBRIDSUS-3564 마스터와 서버 프로세스 간 프로토콜 변경 및 두 개 버전을 동시에 운영하는 경우 포트 설정 필요.....	33
CUBRIDSUS-2828 데이터베이스 이름에 @를 포함할 수 없음.....	33
CUBRIDSUS-3267 Windows 환경에서 디렉토리 경로 설정 시 주의 사항.....	33
CUBRIDSUS-3553 CUBRID 소스 빌드 후 실행 시, 매니저 서버 프로세스 관련 오류 발생.....	33
CUBRID 매니저에서 [볼륨 자동 추가 기능 사용] 옵션을 선택하고 DB를 생성하는 경우 주의 사항.....	33

1. 개요

릴리스 노트 정보

이 문서는 CUBRID 2008 R3.0 Beta 버전에 관한 유용한 정보를 포함한다. 릴리스 노트의 최신 버전은 CUBRID 오픈 소스 프로젝트 사이트(<http://dev.naver.com/projects/cubrid>)에서 확인할 수 있다.

※ 이하 **CUBRID 2008 R3.0 Beta**는 **CUBRID 2008 R3.0**로 표기한다.

릴리스 노트 개정 내역

CUBRID 2008 R3.0 버전의 릴리스 이후 릴리스 노트의 변경 사항은 아래와 같다.

작성 날짜	설명
2010년 7월	CUBRID 2008 R3.0 Beta 릴리스

참고 문서

CUBRID 2008 R3.0 제품과 함께 배포되는 문서는 아래와 같다.

문서	설명
릴리스 노트	CUBRID 릴리스 버전의 특징 및 이전 버전과 변경된 사항과 관련된 정보를 포함한다.
매뉴얼	Quick Start Guide, CUBRID Architecture, SQL 설명서, 튜닝 안내서, 관리자 안내서, CUBRID 매니저 안내서, API References를 포함한다.

버그 리포트 및 사용자 피드백 제공 방법

CUBRID 프로젝트에서는 사용자의 거침없는 버그 리포트와 솔직한 피드백을 기다리고 있으며, 아래 사이트에서 등록할 수 있다.

문서	설명
버그 리포트	CUBRID 오픈 소스 프로젝트: http://dev.naver.com/projects/cubrid/issue
사용자 피드백	CUBRID 오픈 소스 프로젝트: http://dev.naver.com/projects/cubrid/forum CUBRID 공식 사이트: http://www.cubrid.com

추가 정보 안내

CUBRID에 관한 유용한 정보는 아래 사이트에서 찾을 수 있다.

정보	사이트
CUBRID 제품 정보	http://cubrid.com/zbxe/product
CUBRID 라이선스 정보	http://cubrid.com/zbxe/bbs_oss_guide/32249
CUBRID 사용자 문서	http://cubrid.com/zbxe/developer
CUBRID 교육 서비스	http://cubrid.com/zbxe/education_overview

2. CUBRID 2008 R3.0 정보

CUBRID 2008 R3.0 릴리스 특징

CUBRID 2008 R3.0 릴리스는 CUBRID 기반의 응용 프로그램을 보다 쉽고 편리하게 작성할 수 있도록 다양한 연산자, 함수 및 구문을 확장하였다. 또한, 서비스 운영 편의성을 향상시키기 위하여 **cubrid compactdb** 유틸리티를 개선하고 CUBRID HA 기능 안정화에 주력하였다.

- SQL 문법 확장 지원

- **CREATE** 문 확장: 기존 테이블의 스키마 복제, 데이터 전체 복제를 통한 테이블 생성 기능 지원
- **INSERT** 문 확장: 멀티 레코드 입력 가능, **REPLACE** 문 지원
- **SELECT** 문 확장: **LIMIT** 절 지원, **GROUP BY** 절 문법 확장 지원
- **ALTER** 문 확장: 추가할 컬럼의 위치 지정 기능, 인덱스 및 제약 조건 추가/삭제 기능 확장 지원
- 그 외, **TRUNCATE** 문, prepared statement 문 등을 지원

- 연산자 및 함수 추가 지원

- 논리 연산자, 비교 연산자, 비트 연산자 지원
- 다양한 수학 함수, 날짜/시간 데이터 함수 및 날짜 출력 포맷 지정자 추가 지원
- 다양한 문자열 함수를 추가 지원
- PHP API에 20여 개 함수 추가 지원

- 운영 편의성 개선

- CUBRID HA 기능 안정화
- 데이터베이스 공간 정리 작업을 온라인에서 수행할 수 있도록 **cubrid compactdb** 유틸리티에 옵션 추가 지원

보다 자세한 내용은 아래의 [CUBRID 2008 R3.0에서 변경된 사항](#)을 참고한다.

지원 플랫폼 및 설치 권장 사양

CUBRID 2008 R3.0 버전과 호환되는 플랫폼과 설치를 위한 하드웨어/소프트웨어 요구 사항은 아래 표와 같다.

지원 플랫폼	메모리 크기	디스크 공간	필요 소프트웨어
Windows 계열 32/64 Bit (XP, 2003, Vista)	1G 이상	500MB 이상	- JRE 1.5 이상 (CUBRID 매니저 또는 Java 저장 프로시저를 사용하는 경우 필요) - .NET Framework 2.0 이상 (CUBRID Service Tray를 사용하는 경우에 필요) - Microsoft Visual C++ 2008 재배포 가능 패키지
Linux 계열 32/64 Bit (Linux kernel 2.4 및 glibc 2.3.4 이상)	1G 이상	500MB 이상	- JRE 1.5 이상 (CUBRID 매니저 또는 Java 저장 프로시저를 사용하는 경우 필요)

Microsoft Visual C++ 2008 재배포 가능 패키지는 아래의 링크에서 다운로드 가능하다. (언어 확인 필요)

x86용 <http://www.microsoft.com/downloads/details.aspx?familyid=9b2da534-3e03-4391-8a4d-074b9f2bc1bf>

라이선스 안내

CUBRID의 서버 엔진에는 GNU GPL v2 or later 가 적용되고 CUBRID 매니저 및 인터페이스(API)에는 BSD 라이선스가 적용된다. 보다 상세한 정보는 CUBRID 공식 사이트의 [라이선스 가이드](#)를 참고한다.

버전 호환성과 운용성

응용 프로그램 및 CUBRID 매니저의 호환성

- 이전 버전의 JDBC, PHP, CCI, C API 등을 사용하는 응용 프로그램은 CUBRID 2008 R3.0 데이터베이스에 접근할 수 있다. 다만, JDBC, CCI, PHP 인터페이스에 추가/개선된 기능을 사용하기 위해서는 CUBRID 2008 R3.0 버전의 라이브러리를 링크하여야 한다.
- CUBRID 매니저는 CUBRID 2008 R2.1 이상 버전의 서버에 대해서 하위 호환성을 보장하며, 각 서버 버전과 일치하는 CUBRID JDBC 라이브러리를 링크하여야 한다.
- 새로운 예약어 추가 및 일부 질의에 대한 스펙 변경으로 인해 질의 결과가 이전 버전과 다를 수 있으므로 주의한다. 보다 상세한 사항은 [업그레이드 주의 사항](#)을 참고한다.

데이터베이스 호환성

서버 버전을 업그레이드하는 경우, 이전 버전 데이터베이스의 마이그레이션 작업을 수행하여야 한다. 보다 상세한 사항은 [데이터베이스 마이그레이션 절차](#)를 참고한다.

상호 운용성

CUBRID DB 서버와 브로커 서버를 분리하여 운영하는 경우, 서버 장비의 운영 체제가 다르더라도 상호 운용성을 보장한다. 단, CUBRID 버전과 운영 체제의 Bit 버전은 서로 동일하여야 한다. 예를 들어, Linux용 64Bit 버전 DB 서버는 Windows용 64Bit 버전 브로커 서버와 상호 운용이 가능하지만, 32Bit 버전 브로커 서버와는 상호 운용이 불가능하다.

CUBRID 2008 R3.0의 설치 방법

Linux에서 설치

Linux용 설치 패키지는 Linux RPM, tar.gz, 바이너리를 포함하는 스크립트로 제공되며, 설치 방법은 [CUBRID 시작> 설치와 실행> Linux에서의 설치와 실행](#)을 참고한다.

Windows에서 설치

Windows용 설치 파일이 제공되며, 설치 마법사를 이용하여 편리하게 설치할 수 있다. 설치 방법은 [CUBRID 시작> 설치와 실행> Windows에서의 설치와 실행](#)을 참고한다.

CUBRID 환경 변수 및 OS 환경 변수 설정

CUBRID를 사용하기 위해서는 CUBRID 환경 변수와 관련 OS 환경 변수를 설정하여야 한다. 특히, CUBRID 매니저와 Java 저장 프로시저를 사용하기 위해서는 Java 환경 변수를 설정하여야 한다. 설정 방법은 [CUBRID 시작> 꼭 알아두어야 할 것> 환경 변수 설정](#) 및 [SQL 설명서> Java 저장 함수/프로시저> 환경 설정](#)을 참고한다.

CUBRID 2008 R3.0 버전으로 업그레이드하는 방법

업그레이드 주의 사항

● 추가된 예약어 및 허용되지 않는 식별자 확인

SQL 구문 확장 및 함수/연산자 추가 지원으로 인해 아래의 예약어가 추가되었으며, 식별자에 포함할 수 없는 특수 문자가 변경되었다. 이전 버전에서 이러한 식별자가 사용된 경우라면, 업그레이드 이후 오류가 발생하게 되므로 주의하여야 한다. 보다 자세한 사항은 [데이터베이스 마이그레이션 절차](#), [SQL 설명서> 예약어](#) 및 [SQL 설명서> 식별자](#)를 참고한다.

항목	CUBRID 2008 R3.0에서 허용하지 않는 식별자
추가된 예약어	DATABASE, DAY_MILLISECOND, DAY_SECOND, DAY_MINUTE, DAY_HOUR, DISTINCTROW, DIV, DO, DUPLICATE, HOUR_MILLISECOND, HOUR_SECOND, HOUR_MINUTE, LOCALTIME, LOCALTIMESTAMP, MINUTE_MILLISECOND, MINUTE_SECOND, MOD, ROLLUP, SECOND_MILLISECOND, TRUNCATE, XOR, YEAR_MONTH
예약어를 식별자로 사용하기 위한 처리 부호	큰따옴표("), 대괄호([]), 그레이브 부호(`)
식별자에 포함할 수 없는 특수 문자	%, !, , ^, ~

● 응용 프로그램의 수정 필요 여부 검토

일부 함수 및 SQL 질의 수행 방식이 변경되어 질의 결과가 이전 버전과 다를 수 있으므로 주의하여야 한다. 이전 버전과 다른 결과를 반환하는 질의는 다음과 같다.

이전과 다른 결과를 반환하는 질의 유형	이전 버전	CUBRID 2008 R3.0	상세 정보
AUTO_INCREMENT 컬럼에 NULL 입력되는 경우	NULL 저장	자동 증가 값 저장	바로가기
외래 키에서 ON UPDATE/ON DELETE ACTION으로 NO ACTION이 정의된 경우	아무 동작하지 않음	RESTRICT와 동일하게 동작	바로가기
POWER 함수의 반환 값 타입	인자 타입에 따라 다름	DOUBLE 타입	바로가기
TO_CHAR 함수에서 'HH' 또는 'HH12' 지정자로 12시간 포맷을 사용하는 경우	표현 범위: 0~11	표현 범위: 1~12	바로가기
두 자리 수 연도 값의 변환 결과	01~99 → 0001 ~ 0099	00~69 → 2000~2069 70~99 → 1970~1999	바로가기
날짜/시간 데이터 타입의 비교/산술 연산	동일한 타입에 대해서만 비교 가능 동일한 타입에 대해서만 뺄셈 가능	DATE, TIMESTAMP, DATETIME 타입 간 비교 가능 DATE, TIMESTAMP, DATETIME 타입 간 뺄셈 가능	바로가기
JDBC 에서 URL String 입력 정보	속성(PROPERTY) 정보 앞에 물음표 생략 가능	속성(PROPERTY) 정보 앞에 반드시 물음표 명시	바로가기

● 파라미터 변경 사항 확인

아래의 파라미터는 변경 사항이 있으므로, 환경 설정 파일을 적용하는 경우 주의한다.

파라미터 이름	변경 사항	상세 정보
INSERT_EXECUTION_MODE	파라미터 값 설정 범위 확장	바로가기
ANSI_QUOTES	추가됨 (디폴트 값 yes)	바로가기
PIPES_AS_CONCAT	추가됨 (디폴트 값 yes)	
ONLY_FULL_GROUP_BY	추가됨 (디폴트 값 no)	

● 기존 환경 설정 파일 보관

이전 버전의 \$CUBRID/conf 디렉터리의 환경 설정 파일(**cubrid.conf**, **cubrid_broker.conf**, **cm.conf**)과 \$CUBRID/DATABASES 디렉터리의 데이터베이스 위치 정보 파일(**databases.txt**)을 보관한다.

● 데이터베이스 마이그레이션

CUBRID 2008 R3.0 버전은 하위 버전의 데이터베이스와 호환되지 않으므로, 기존 데이터베이스를 CUBRID 2008 R3.0 버전으로 마이그레이션 하여야 한다. ([데이터베이스 마이그레이션 절차 참고](#))

● 복제 또는 HA 환경 재구성

이전 버전의 복제 기능을 사용하는 시스템에서는 보다 안정적인 운영을 위해 DB 마이그레이션 및 HA 환경으로 재구성할 것을 권장한다. 또한, CUBRID 2008 R2.2 이전 버전에서 제공된 Linux Heartbeat 기반의 HA 기능을 사용하는 시스템도 보다 안정적인 운영을 위해 DB 마이그레이션 및 CUBRID Heartbeat 기반의 HA 환경으로 재구성할 것을 권장한다. ([HA 환경에서 마이그레이션 절차 참고](#))

데이터베이스 마이그레이션 절차

<CUBRID 설치 디렉토리>/bin/migrate_r30 유틸리티를 이용하여 마이그레이션을 수행할 수 있으며, 아래의 수행 가이드를 참고한다. 이때, 다운로드 페이지에서 별도 배포되는 **check_reserved.sql** 예약어 검출 스크립트를 이용하여 기존 버전의 데이터베이스에 사용된 식별자 중 CUBRID 2008 R3.0에서 추가된 예약어가 존재하는지를 검출할 수 있다.

다른 방법으로는 **cubrid unloaddb/loaddb** 유틸리티를 사용하여 마이그레이션을 수행할 수 있으며, 이는 [관리자 안내서> 데이터베이스 마이그레이션](#)을 참고한다.

단계	Linux 환경	Windows 환경
C1 단계: CUBRID Service 종료	% cubrid service stop	CUBRID 서비스 트레이>[Exit] 선택
C2 단계: 예약어 검출 스크립트 실행	예약어 검출 스크립트가 위치하는 디렉토리에서 아래 명령을 실행한다. 검출 결과를 확인하여 마이그레이션 진행 또는 식별자 수정 작업을 진행한다. SQL 설명서> 식별자 를 참고한다. % csql -S -u dba -i check_reserved.sql <r22 db name>	
C3 단계: 이전 버전 DB 백업	2008 R3.0 운영 중 이전 버전으로 복구하는 상황에 대비하기 위해 백업을 수행하고, 백업 파일을 별도 디렉토리(R22_backup)에 보관한다. (C2a) % mkdir R22_backup % cubrid backupdb -S -D R22_backup <r22_db_name> 이전 버전의 databases.txt 및 conf 디렉토리 내 설정 파일을 별도 디렉토리에 보관한다. (C2b)	
		이전 버전 CUBRID를 제거한다. 주의할 점은, 이전 버전의 데이터베이스가 삭제되지 않도록 "데이터베이스와 설정 파일을 모두 삭제하시겠습니까?"라는 요청 창에서 반드시 "아니오"를 선택한다.
C4 단계: 2008 R3.0 버전 설치	본 문서의 CUBRID 2008 R3.0의 설치 방법 을 참고한다.	
C5 단계: 마이그레이션 도구 실행		CUBRID 서비스 트레이> [CUBRID Server]>[Stop] 선택하여 서버 종료

	(C2b)에서 보관한 databases.txt를 2008 R3.0의 설치 디렉터리에 복사한다. (C5a) 아래와 같이 migrate_r30 유틸리티를 실행한다. (C5b) <code>% migrate_r30 <db_name></code>	
C6 단계: 2008 R3.0 버전 DB 백업	데이터베이스 버전이 R1.x인 경우에만 수행한다. 이전 버전이 R2.x 버전인 경우, 이 단계를 생략할 수 있다. <code>% cubrid backupdb -S <db_name></code>	
C7 단계: CUBRID 환경 설정 및 CUBRID Service 구동	환경 설정 파일을 수정한다. 이때, (C2b)에서 보관한 이전 버전의 환경 설정 파일을 사용할 수 있다. <code>% cubrid service start</code> <code>% cubrid server start <db_name></code>	CUBRID 서비스 트레이> [CUBRID Server]-> [Start]

HA 환경에서 데이터베이스 마이그레이션 절차

CUBRID 2008 R2.0, R2.1 버전의 HA 기능을 사용하는 경우, 해당 버전에서 사용되었던 Linux Heartbeat 패키지가 더 이상 사용되지 않으므로 아래 가이드를 참고하여 서버 버전 업그레이드, DB 마이그레이션을 수행한 후 HA 환경을 새롭게 구축하여야 한다. 한편, CUBRID 2008 R2.2 이상 버전의 HA 기능을 사용하는 경우에는 서버 업그레이드, DB 마이그레이션만 수행하면 된다.

아래는 브로커, 마스터 DB, 슬레이브 DB를 각각 별도 서버에 구축한 환경에서 현재 서비스를 중지하고 업그레이드를 수행하기 위한 가이드이다. 서비스 무정지 업그레이드 시나리오는 별도 가이드 문서를 참고한다.

단계	설명
H1 단계: HA 관련 서비스 종료 및 기존 Linux heartbeat 제거	<code>% cubrid broker stop</code> <code>% service heartbeat stop</code> <code>% chkconfig --del heartbeat</code> <code>% pkill -u user1 -f "cub_master"</code>
H2~H6 단계: 마스터 서버에서 C2~C6 단계를 수행	마스터 서버에서 CUBRID 업그레이드 및 데이터베이스 마이그레이션을 수행하고, 2008 R3.0 데이터베이스를 백업한다.
H7 단계: 슬레이브 서버에 CUBRID 2008 R3.0 버전 설치	설치 방법은 본 문서의 CUBRID 2008 R3.0의 설치 방법 을 참고한다.
H8 단계: 마스터 백업본을 슬레이브 서버에서 복구	H6에서 생성된 마스터 서버의 2008 R3.0 데이터베이스 백업본(testdb_bk*)을 슬레이브 서버에서 복구한다. <code>% scp user1@m_server:~/DB/testdb/testdb_bk0v000 .</code> <code>% scp user1@m_server:~/DB/testdb/log/testdb_bkvinf ./log/.</code> <code>% cubrid restoredb testdb</code>
H9 단계: HA 환경 재구성 후 HA 모드 구동	HA 구동 스크립트(cubrid-ha) 및 환경 설정 파일(cubrid.conf)을 설정한다. 관리자 안내서> CUBRID HA 환경 설정 을 참고한다. 마스터 서버 및 슬레이브 서버에서 HA 모드로 DB를 구동한다. 관리자 안내서> CUBRID HA 모드 구동 을 참고한다. <code>[root@m_server~]# service cubrid-ha start</code> <code>[root@s_server~]# service cubrid-ha start</code>
H10 단계: 브로커 서버에 CUBRID 2008 R3.0 버전 설치 및 브로커 구동	설치 방법은 본 문서의 CUBRID 2008 R3.0의 설치 방법 을 참고한다. 브로커 설정 후 브로커를 시작한다. 관리자 안내서> CUBRID HA 환경 설정 을 참고한다. <code>% cat cubrid broker.conf</code> ... <code>ACCESS_MODE=RW</code> <code>% cubrid broker start</code>

복제 재구성 방법

- 복제 기능을 사용 중인 환경에서는 보다 안정적인 운영을 위하여 HA 기능을 사용하여 이중화 환경을 구축하는 것을 권장한다. 서버 버전 업그레이드 및 데이터베이스 마이그레이션을 수행한 후, HA 환경을 새롭게 구축할 수 있다.
- 복제 기능에 관한 개발은 향후 중단될 것이다(Outdated and Deprecated Feature).

3. CUBRID 2008 R3.0 버전에서 변경된 사항

새로 추가된 기능 - SQL 문법 확장 관련

CUBRIDSUS-3590 CREATE TABLE 문 확장 및 테이블 복제 기능 지원

- 기존 테이블과 동일한 스키마를 가지는 테이블을 생성하는 기능 지원

기존 테이블의 스키마만 복제하여 새로운 테이블을 생성할 수 있도록 **CREATE TABLE** 문법을 확장하였다. **CREATE TABLE ... LIKE** 문을 사용하여 보다 편리하게 새로운 테이블의 스키마를 정의할 수 있다.

```
// a_tbl 테이블의 스키마만 복제하여 새로운 테이블 생성
CREATE TABLE new_tbl LIKE a_tbl;
```

- 관련 문서: [테이블 정의> CREATE TABLE LIKE](#)

- 기존 테이블에 대한 **SELECT** 부질의 결과 레코드를 포함하는 테이블을 생성하는 기능 지원

다수 개의 레코드가 삽입된 테이블이 존재하는 경우, **SELECT** 문의 결과 레코드를 포함하는 새로운 테이블을 생성할 수 있도록 **CREATE TABLE** 문법을 확장하였다. **CREATE TABLE ... AS SELECT** 문을 사용하여 보다 편리하게 새로운 테이블의 스키마 정의 및 레코드 삽입을 수행할 수 있다.

```
// a_tbl 테이블에 대한 SELECT 결과를 복제하여 새로운 테이블 생성
CREATE TABLE new_tbl1 AS SELECT * FROM a_tbl;
```

- 관련 문서: [테이블 정의> CREATE TABLE AS SELECT](#)

- 테이블 생성 시 인덱스 생성 기능 지원

테이블 생성 시 컬럼 정의문 내에서 **INDEX**, **KEY**, **UNIQUE INDEX** 를 정의하여 인덱스를 생성할 수 있도록 **CREATE TABLE** 문법을 확장하였다. 그리고, 인덱스 정의 시 대상 컬럼 이름 뒤에 **ASC** 또는 **DESC** 옵션을 명시할 수 있으며, **GROUP BY** 절 내의 컬럼에 대해서도 컬럼 옵션을 명시하여 간편하게 정렬을 수행할 수 있다.

```
// 테이블 생성 시 인덱스 정의
CREATE TABLE new_tbl(
  id INT,
  name VARCHAR,
  CONSTRAINT INDEX(id DESC, name ASC)
);
```

- 관련 문서: [테이블 정의> CREATE TABLE, 컬럼 옵션](#)

CUBRIDSUS-3590 prefix_length에 해당하는 앞 부분 스트링에 대해

인덱스 생성 기능 지원 및 시스템 카탈로그 테이블 변경

문자열 타입 컬럼에 인덱스를 정의하는 경우, **prefix_length** 를 명시하여 스트링의 앞 부분(prefix) 일부에 대해서만 인덱스를 생성할 수 있다. 이 기능과 관련하여 시스템 카탈로그 가상 테이블 **DB_INDEX_KEY** 에 **prefix_length** 값이 저장되는 **key_prefix_length** 컬럼이 추가되었다.

```
CREATE TABLE new_tbl ( name VARCHAR(20));
INSERT INTO new_tbl VALUES ('abc1234'), ('abcd123');
// 스트링 타입 컬럼에 대해 3 바이트 길이 prefix에 대해 인덱스 생성
CREATE INDEX ON new_tbl(name(3));

// 시스템 카탈로그에서 정의된 prefix_length 값 조회
SELECT class name, key attr name, key prefix length FROM db index key
WHERE class_name = 'new_tbl';
```

- 관련 문서: [인덱스 정의 > CREATE INDEX](#), [시스템 카탈로그 가상 클래스 > DB_INDEX_KEY](#)

CUBRIDSUS-3590 INSERT 문법 확장 및 REPLACE 문 지원

- 여러 개의 레코드를 한번에 삽입할 수 있도록 INSERT 문법 확장

INSERT 문에서 INTO 를 생략할 수 있으며, VALUES 대신 VALUE 를 사용할 수도 있다. 또한, 삽입하고자 하는 값을 콤마로 구분하여 여러 행을 한번에 삽입할 수 있도록 구문을 확장하였다.

```
CREATE TABLE insert_tbl (A INT DEFAULT 1, B INT UNIQUE);
INSERT INTO insert_tbl VALUES (1,1), (2,2), (NULL, 3);
```

- 디폴트 값을 간편하게 삽입할 수 있도록 구문 확장

DEFAULT 키워드를 VALUES 앞 또는 뒤에 명시하거나, 컬럼 값에 직접 명시할 수 있도록 구문을 확장하였다.

```
// 정의된 디폴트 값 삽입
INSERT INTO insert_tbl VALUES DEFAULT;
INSERT INTO insert_tbl DEFAULT VALUES;

// 컬럼 값에 DEFAULT 키워드 직접 명시
INSERT INTO insert_tbl VALUES (DEFAULT, 4), (DEFAULT, 5);
```

- 특정 컬럼 값을 간편하게 삽입할 수 있도록 SET 절 지원

INSERT 문에서 VALUES 절 대신 SET 절을 사용하여 삽입하고자 하는 특정 컬럼 이름 및 컬럼 값을 지정할 수 있다.

```
// 일부 컬럼 값만 삽입
INSERT INTO insert_tbl SET A=10;
```

- UNIQUE 컬럼에 중복된 값을 삽입할 수 있도록 ON DUPLICATE KEY UPDATE 절 지원

INSERT 문에서 ON DUPLICATE KEY UPDATE 절을 사용하여 UNIQUE 또는 PRIMARY KEY 제약 조건을 위반하는 레코드가 삽입되더라도 컬럼 값을 새로운 값으로 갱신할 수 있다.

```
// 중복된 값 삽입 시 에러 출력 없이 명시된 컬럼 값으로 갱신됨
INSERT INTO insert_tbl VALUES (DEFAULT, 5) ON DUPLICATE KEY UPDATE A=2;
```

- 레코드 삭제 후 삽입을 수행하는 REPLACE 문 추가 지원

INSERT 문과 유사하지만, ON DUPLICATE KEY UPDATE 절을 명시하지 않고도 UNIQUE 또는 PRIMARY KEY 제약 조건을 위반하는 레코드가 삽입되더라도 해당 컬럼 값을 새로운 값으로 갱신할 수 있는 REPLACE 문을 지원한다.

```
// 중복된 값 삽입 시 기존 레코드 삭제 후 새로운 레코드가 삽입됨
REPLACE INTO insert_tbl VALUES (3, 5);
```

- 관련 문서: [데이터 조회 및 조작 > INSERT](#), [ON DUPLICATE KEY UPDATE 절](#), [REPLACE](#)

CUBRIDSUS-3590 SELECT 문, DELETE 문, UPDATE 문에서 대상 레코드의 개수를 제한하는 LIMIT 절 지원

LIMIT 절을 사용하여 SELECT 문에서 실행 결과 셋의 레코드 개수를 제한할 수 있다. 마찬가지로 DELETE 문 및 UPDATE 문에서도 연산을 수행할 레코드의 개수를 제한할 수 있다.

```
//결과 셋의 2 번째 레코드부터 3 개만 출력
SELECT * FROM a_tbl ORDER BY a LIMIT 2,3;

//조건을 만족하는 레코드 중 처음 3 개만 업데이트
UPDATE a_tbl SET a=10, b=10 WHERE a IS NULL LIMIT 3;
```

```
//조건을 만족하는 레코드 중 처음 1 개만 삭제
DELETE FROM a_tbl WHERE a IS NULL LIMIT 1;
```

- 관련 문서: [데이터 조회 및 조작 > SELECT > LIMIT 절](#)

CUBRIDSUS-3590 SELECT 문에서 FROM 절 생략 가능

SELECT 문에서 참조하는 테이블이 없는 경우 이를 생략할 수 있도록 문법을 확장하였다.

```
SELECT 1+1;
SELECT FALSE;
```

- 관련 문서: [데이터 조회 및 조작 > SELECT > FROM 절](#)

CUBRIDSUS-3590 SELECT 문에서 GROUP BY ... HAVING 절의 문법 확장

지원

GROUP BY 절에서 컬럼 이름 대신 컬럼 별칭을 명시할 수 있으며, 그룹 별 중간 집계 및 총 집계를 출력할 수 있도록 **WITH ROLLUP** 구문을 지원한다. 또한, **ORDER BY NULL** 구문을 명시하여 **GROUP BY** 절에 의한 정렬이 수행되지 않도록 할 수 있다.

한편, **ONLY_FULL_GROUP_BY** 파라미터 설정 값이 NO인 경우에는, **GROUP BY** 절에 명시하지 않은 컬럼을 **SELECT** 리스트에서 사용할 수 있도록 확장된 문법을 제공한다.

```
//정렬 작업이 수행되지 않도록 ORDER BY NULL 을 명시
SELECT dept no, avg(sales amount) FROM sales tbl
GROUP BY dept_no ORDER BY NULL;
```

```
//컬럼 별칭을 명시하고, 그룹별 합계 및 총 합계가 출력되도록 WITH ROLLUP 을 명시
SELECT dept no AS a1, name AS a2, avg(sales amount) AS a3 FROM sales tbl
GROUP BY a1,a2 WITH ROLLUP;
```

- 관련 문서: [데이터 조회 및 조작 > SELECT > GROUP BY HAVING 절](#)

CUBRIDSUS-3590 ALTER TABLE 문법 확장 및 추가할 컬럼 위치 지정

기능 지원

- 인덱스 및 제약 조건 추가 및 삭제 기능 지원

ALTER TABLE 문을 사용하여 보다 간편하게 인덱스 및 제약 조건 정의를 변경할 수 있도록 **ADD INDEX** 절, **DROP INDEX** 절, **DROP PRIMARY KEY** 절, **DROP FOREIGN KEY** 절을 추가 지원한다.

```
// ALTER TABLE 문 내에서 인덱스 추가
ALTER TABLE a_tbl ADD INDEX (a ASC), ADD INDEX (b DESC);
```

```
// ALTER TABLE 문 내에서 인덱스 삭제
ALTER TABLE a_tbl DROP INDEX i a_tbl a;
```

```
// ALTER TABLE 문 내에서 기본 키 삭제
ALTER TABLE a_tbl DROP PRIMARY KEY;
```

```
// ALTER TABLE 문 내에서 외래 키 삭제
ALTER TABLE a_tbl DROP FOREIGN KEY fk_a_tbl_a;
```

- 추가할 컬럼 위치 지정 기능 추가

ALTER TABLE ... ADD COLUMN 절을 사용하여 새로운 컬럼을 추가하는 경우, **FIRST** 또는 **AFTER** 키워드를 사용하여 해당 컬럼의 위치를 지정할 수 있는 기능을 지원한다.

```
CREATE TABLE a_tbl(a INT, b INT);
INSERT INTO a_tbl VALUES(1, 1),(2, 2);
```

```
// ALTER TABLE 문에서 컬럼 추가 시 위치 지정
ALTER TABLE a_tbl ADD COLUMN c INT DEFAULT 0 AFTER a;
```

- 관련 문서: [테이블 정의> ALTER TABLE](#)

CUBRIDSUS-3590 TRUNCATE 문을 사용한 모든 레코드 삭제 기능 지원

TRUNCATE 문을 사용하여 테이블 내 모든 레코드를 삭제할 수 있으며, 이는 대상 테이블에 정의된 모든 인덱스와 제약 조건을 먼저 삭제한 후 레코드를 삭제하므로, 기존 **DELETE** 문을 사용하는 경우보다 성능 상 유리하다.

```
CREATE TABLE b_tbl(A INT PRIMARY KEY);
INSERT INTO b_tbl VALUES (1),(2),(3);
TRUNCATE b_tbl;
```

- 관련 문서: [데이터 조회 및 조작> TRUNCATE](#)

CUBRIDSUS-3590 외래 키 정의 시 ON UPDATE 및 ON DELETE 연산에 대한 트리거 동작 추가

기본 키 값이 업데이트 또는 삭제되는 경우, 이를 참조하는 외래 키 값을 **NULL**로 갱신하는 **SET NULL** 동작 옵션을 추가 지원한다. 한편, **NO ACTION** 동작 옵션의 경우 이전 버전에서는 기본 키 값의 변경을 허용하고 외래 키에 대해서는 아무런 동작을 수행하지 않았으나, 이를 **RESTRICT** 동작 옵션과 동일하게 동작하도록 수정하였다. 즉, **NO ACTION** 또는 **RESTRICT** 동작 옵션이 정의된 경우, 외래 키가 참조하는 기본 키 값을 변경할 수 없다.

```
<referential_triggered_action_on_update> ::=
ON UPDATE { RESTRICT | NO ACTION | SET NULL }

<referential_triggered_action_on_delete> ::=
ON DELETE { CASCADE | RESTRICT | NO ACTION | SET NULL }
```

- 관련 문서: [테이블 정의> CREATE TABLE> FOREIGN KEY 제약 조건](#)

CUBRIDSUS-3590 SQL 레벨에서 PREPARED STATEMENT 실행 기능 지원

미리 SQL 문을 선언하는 **PREPARE** 문, 이를 실행하는 **EXECUTE** 문, 선언된 SQL 문을 삭제하는 **DROP PREPARE** 문을 지원하며, 이를 사용하여 prepared statement 기능을 사용할 수 있다. 단, CUBRID 매니저의 질의 편집기에서는 이 기능이 지원되지 않으며, CSQL 인터프리터에서는 **PREPARE** 문과 **EXECUTE** 문을 함께 작성한 후 실행 명령어(**;<xr 또는 ;run**)를 입력하여야 한다.

```
PREPARE stmt1 FROM 'SELECT POWER(?,2)*PI()';
EXECUTE stmt1 USING 2;
;run
```

- 관련 문서: [데이터 조회 및 조작> PREPARED STATEMENT](#)

CUBRIDSUS-3590 SQL 확장 문법 적용을 위한 파라미터 추가 지원

SQL의 확장 문법 또는 표준 문법 적용을 위한 파라미터를 추가로 지원한다. 응용 프로그램에서 작성하고자 하는 SQL 유형 및 코딩 스타일에 따라 해당하는 파라미터를 설정할 수 있다.

추가된 파라미터	파라미터 값(디폴트 설정)	파라미터 값
ONLY_FULL_GROUP_BY	NO	YES
	확장 문법 적용.	SQL 표준 문법을 적용.
	GROUP BY 절에 명시되지 않은 컬럼을 SELECT GROUP BY 절에 명시한 컬럼만 SELECT 컬럼 리스트에 명시 가능	컬럼 리스트에 명시 가능
ANSI_QUOTES	YES	NO

	작은 따옴표(')를 문자열 처리 부호로 사용	큰 따옴표(")도 문자열 처리 부호로 사용
PIPES_AS_CONCAT	YES	NO
	이중 파이프 부호()를 문자열 병합 연산자로 사용	이중 파이프 부호()를 부울린 연산자 OR 로 사용

- 관련 문서: [데이터베이스 설정 > 구문/타입 파라미터](#)

새로 추가된 기능 - 연산자와 함수 관련

CUBRIDSUS-3591 논리 연산자 확장 지원

부울린(Boolean) 연산식에 대한 논리 연산자를 확장 지원한다.

논리 연산자	비고
&&	AND 와 동일하게 사용 가능
	OR 와 동일하게 사용 가능 (단, PIPES_AS_CONCAT=NO 파라미터 설정 필요)
XOR	XOR 신규 지원
!	NOT 과 동일하게 사용 가능

- 관련 문서: [연산자와 함수 > 논리 연산자](#)

CUBRIDSUS-3591 비교 연산자 확장 지원

비교 연산자를 확장 지원한다.

비교 연산자	비고
<=>	NULL Safe 등호 신규 지원
!=	<>와 동일하게 사용 가능
IS, IS NOT	조건식에서 연산식과 불리언 값(TRUE, FALSE, UNKNOWN, NULL)의 비교 연산 가능

- 관련 문서: [연산자와 함수 > 비교 연산자](#)

CUBRIDSUS-3591 비트 연산자 및 비트 함수 신규 지원

비트열 리터럴을 확장하며, 비트 연산자 및 비트 연산 함수를 신규 지원한다.

지원 항목	비고
비트열 리터럴	2 진수 형식(B'1010', 0b1010) 및 16 진수 형식(X'a', 0xa) 사용 가능
비트 연산자	&, , ^, ~, <<, >> 신규 지원
비트 함수	BIT_AND(expr), BIT_OR(expr), BIT_XOR(expr), BIT_COUNT(expr) 신규 지원

- 관련 문서: [연산자와 함수 > 비트 함수와 연산자](#)

CUBRIDSUS-3591 수치 연산 함수의 확장 지원

- 삼각 함수의 추가 지원

다양한 삼각 함수를 이용하여 라디안 값을 구할 수 있으며, 변환 함수를 이용하여 값의 단위를 라디안으로 변환 할 수도 있다. 이를 위해 새로 추가된 함수는 다음과 같다.

```
//삼각 함수를 이용하여 라디안 값을 반환
COS( X )
COT( X )
SIN( X )
TAN( X )

//역 삼각 함수를 이용하여 라디안 값을 반환
ACOS( X )
ASIN( X )
ATAN(X[, Y]), ATAN2(X[, Y])

//라디안 단위인 x를 각도로 환산하여 반환
DEGRESS( X )

// 각도 값 x를 라디안 단위 값으로 환산
RADIANS( X )

// PI 값을 DOUBLE로 반환
PI()
```

- 수학 함수의 추가 지원

절대 값, 거듭제곱 값, 제곱근 및 로그 연산을 위해 새로 추가된 함수는 다음과 같다.

```
//거듭 제곱 값 반환
POW( X )

//제곱근 반환
SQRT( X )

//이진 로그, 자연 로그, 상용 로그 값을 반환
LOG2( X ), LN( X ), LOG10( X )
```

- SEED 값 지정이 가능하도록 랜덤 함수의 기능 확장

CUBRID가 지원하는 모든 랜덤 함수에서 seed 값을 지정할 수 있도록 기능이 확장되었다. 또한, 난수를 발생시키는 범위를 0~2¹⁶에서 0~2³²로 확장하였다.

```
//정수 타입 난수 발생
RAND( [seed] )
RANDOM( [seed] )

//실수 타입 난수 발생
DRANDOM( [seed] )
DRAND( [seed] )
```

- 수치 값을 사용자가 지정한 형식의 문자열로 변환하는 FORMAT() 함수 추가 지원

주어진 숫자를 '#,###,###,#####' 형식의 문자열로 변환하는 함수를 추가 지원하며, 소수부의 표현 자리 수를 인자로 지정할 수 있다.

```
// 주어진 값을 세 자리마다 쉼표로 구분하고 지정한 자릿수만큼 표현
SELECT FORMAT(12000.123456,4);

=====
'12,000.1235'
```

- 관련 문서: [수치 연산 함수](#) > [RANDOM/RAND 함수](#), [FORMAT 함수](#) 등

CUBRIDSUS-3591 날짜/시간 함수의 확장 지원 및 출력 포맷의 다양화

● 날짜/시간 값에 대해 시간 간격 단위를 지정하여 연산을 수행하는 함수 추가 지원

밀리초(MILLISECOND)부터 년(YEAR) 단위까지 다양한 시간 간격 단위를 지정하여 주어진 날짜 값에 특정 시간 간격과의 차이를 연산할 수 있다. 이를 위해 새로 추가된 함수는 다음과 같다.

```
//ADDDATE( ), DATE_ADD( ) 함수를 사용하여 주어진 날짜에서 시간 간격을 더한 값을 반환
SELECT ADDDATE('2010-06-20', INTERVAL 60 SECOND), ADDDATE('2010-06-20', 1);

=====
'12:01:00.000 AM 06/20/2010' '06/21/2010'

//SUBDATE( ), DATE_SUB( ) 함수를 사용하여 주어진 날짜에서 시간 간격을 뺀 값을 반환
SELECT SUBDATE('2010-06-20', INTERVAL 60 SECOND), SUBDATE('2010-06-20', 1);

=====
'11:59:00.000 PM 06/19/2010' '06/19/2010'

// 주어진 날짜 간 차이를 일 단위로 반환
SELECT DATEDIFF(SYSDATETIME, '2010-06-20');

=====
12
```

● 날짜/시간 값을 사용자가 지정한 형식의 문자열로 변환하는 함수의 추가 지원

날짜/시간 값을 사용자가 지정한 형식의 문자열로 변환하는 함수를 추가 지원한다. 또한, 추가로 지원하는 '%'로 시작되는 새로운 포맷 지정자(specifier)를 결합하여 다양한 출력 포맷을 지정할 수 있다.

```
// 주어진 값을 'MM/DD/YYYY' 형식의 문자열을 출력
SELECT DATE(NOW());

=====
'07/02/2010'

// 주어진 값을 지정된 형식의 문자열로 변환 출력
SELECT DATE FORMAT('2010-06-20 22:23:00', '%Y %M %W %H:%i');

=====
'2010 June Sunday 22:23'

// time 값을 지정된 형식의 문자열로 변환 출력
SELECT TIME FORMAT('23:59:01', '%H %h %i %s %f');

=====
'23 11 59 01 000'
```

● 문자열을 날짜/시간 값으로 변환하는 함수의 추가 지원

문자열을 지정한 형식을 적용하여 날짜/시간 값으로 변환하는 함수를 추가로 지원한다. **STR_TO_DATE()** 함수에 적합한 포맷 지정자를 적용하여 문자열을 날짜/시간 값으로 변환할 수 있다. 또한, 문자열을 밀리초 단위의 타임스탬프 값으로 변환하는 **TIMESTAMP()** 함수를 추가로 제공한다.

```
// 문자열을 지정된 날짜/시간 타입으로 변환 출력
SELECT STR_TO_DATE('June 20, 2010', '%M %d,%Y');

=====
06/20/2010

// 문자열을 DATETIME 타입 값으로 반환 또는 시간 값을 더하여 출력
SELECT TIMESTAMP('2010-06-20'), TIMESTAMP('2010-06-20','13:59:59');

=====
12:00:00.000 AM 06/20/2010 01:59:59.000 PM 06/20/2010
```

● 현재 시스템 날짜/시간을 구하는 함수의 추가 지원

시스템의 현재 날짜/시간을 구하는 기존 함수들의 동의어를 추가하였으며, UNIX 타임 스탬프 값을 출력하는 함수를 추가 지원한다.

```
// 현재 날짜를 반환
CURDATE(), CURRENT_DATE(), CURRENT_DATE, SYSDATE, SYS_DATE

// 현재 시간을 반환
CURTIME(), CURRENT_TIME(), CURRENT_TIME, SYSTIME, SYS_TIME

// 현재 타임스탬프 값을 초 단위까지 반환
CURRENT_TIMESTAMP(), CURRENT_TIMESTAMP, LOCALTIME(), LOCALTIME, LOCALTIMESTAMP(), LOCALTIMESTAMP,
SYSTIMESTAMP, SYS_TIMESTAMP

// 현재 타임스탬프 값을 밀리초 단위까지 반환
CURRENT_DATETIME(), CURRENT_DATETIME, NOW(), SYSDATETIME, SYS_DATETIME

// UNIX_TIMESTAMP() 함수를 사용하여 '1970-01-01 00:00:00' UTC 이후 지정된 날짜까지의 초 단위 시간 간격을 반환
SELECT UNIX_TIMESTAMP('2010-06-20')

=====
1276959600
```

- 관련 문서: [날짜/시간 함수와 연산자](#), [ADDDATE 함수](#), [STR_TO_DATE 함수](#), [DATE_FORMAT 함수](#) 등

CUBRIDSUS-3591 문자열 함수의 확장 지원

● 컬럼 값을 연결한 문자열을 가로 출력하는 CONCAT() 함수 지원

컬럼 값을 병합하여 하나의 문자열로 출력하는 **CONCAT()** 함수 및 연결할 값 사이에 구분자를 삽입하여 출력할 수 있는 **CONCAT_WS()** 함수를 함께 제공한다.

```
SELECT CONCAT('CUBRID', '2008', 'R3.0'), CONCAT_WS('/', 'CUBRID', '2008', 'R3.0');

=====
'CUBRID2008R3.0'      'CUBRID//2008//R3.0'
```

● 문자열 비교 및 검색하는 함수 추가 지원

주어진 위치로부터 특정 길이만큼의 문자열을 반환하는 함수로 **MID()** 함수를 추가로 지원하며, 가장 왼쪽 또는 가장 오른쪽 위치로부터 특정 길이만큼의 문자열을 반환하는 **LEFT()** 함수 및 **RIGHT()** 함수도 추가로 제공한다.

```
SELECT MID('12345abcde', 6, 4), LEFT('12345abcde', 4), RIGHT('12345abcde', 4);

=====
'abcd'      '1234'      'bcde'
```

주어진 문자열의 위치를 반환하는 함수로 **LOCATE()** 함수를 추가로 제공한다.

```
SELECT LOCATE('abc', '12345abcde');

=====
6
```

주어진 문자열과 나머지 인자로 지정된 문자열을 비교하여, 동일한 인자의 위치를 반환하는 **FIELD()** 함수를 추가로 제공한다.

```
SELECT FIELD('abc', 'a', 'ab', 'abc', 'abcd');

=====
3
```

주어진 두 개의 문자열을 비교하여 0, 1, -1 을 반환하는 **STRCMP()** 함수를 추가로 제공한다.

```
SELECT STRCMP('abc', 'abc'), STRCMP('abc', 'ab'), STRCMP('abc', 'abcd');

=====
0      1      -1
```

- 문자열을 변환하는 함수 추가 지원

주어진 문자열을 대문자 또는 소문자로 변환하는 **UCASE()** 함수 및 **LCASE()** 함수를 각각 추가 지원한다. 또한, 주어진 문자열을 역순으로 변환하여 반환하는 **REVERSE()** 함수를 추가 지원한다.

```
SELECT UCASE('CUBrid'), LCASE('CUBrid'), REVERSE('CUBrid');

=====
'CUBRID'          'cubrid'          'dirBUC'
```

- 관련 문서: [스트링 함수와 연산자](#)> [CONCAT 함수](#), [MID 함수](#), [LOCATE 함수](#), [REVERSE 함수](#) 등

CUBRIDSUS-3591 정보 함수의 확장 지원

- 데이터베이스 관련 정보를 반환하는 함수 추가 지원

현재 연결된 데이터베이스 이름, 대상 서버에 존재하는 모든 데이터베이스 이름 및 데이터베이스 사용자 이름을 반환하는 정보 함수를 추가 지원한다.

```
//연결된 데이터베이스 이름 반환
SELECT DATABASE(), SCHEMA();

//서버에 존재하는 모든 데이터베이스 이름 반환
SELECT LIST_DBS();

//데이터베이스 사용자 이름 및 호스트 이름을 반환
SELECT USER(), SYSTEM_USER();
```

- 컬럼에 정의된 디폴트 값을 반환하는 함수 추가 지원

주어진 컬럼에 정의된 디폴트 값을 반환하는 **DEFAULT()** 함수를 추가 지원한다.

```
CREATE TABLE info_tbl(id INT DEFAULT 0)
INSERT INTO info_tbl VALUES (1);

//컬럼에 정의된 디폴트 값 반환
SELECT id, DEFAULT(id) FROM info_tbl;

=====
1          0
```

- 이전 질의문에 의해 변경된 행의 개수를 반환하는 함수 추가 지원

이전 질의문에 의해 변경(UPDATE, INSERT, DELETE)된 행의 개수를 반환하는 함수를 추가로 지원한다.

```
INSERT INTO info_tbl VALUES (4), (5), (6);
//변경된 행의 개수를 반환
SELECT ROW_COUNT();
;xr

=====
3
```

- 관련 문서: [정보 함수](#)> [USER 함수](#), [DEFAULT 함수](#), [ROW_COUNT 함수](#) 등

CUBRIDSUS-3591 조건 연산 함수의 확장 지원

- NULL 값 비교를 위한 ISNULL(), IFNULL() 함수 추가 지원

특정 컬럼 값이 NULL 인지 비교하고, 결과에 따라 다른 값을 반환하는 함수를 추가 지원한다.

```
CREATE TABLE case_tbl(a INT);
INSERT INTO case_tbl VALUES (1), (2), (NULL);
//expr 이 NULL 이면 1, NULL 이 아니면 0 을 반환
SELECT ISNULL(a) FROM case_tbl;
```

```
//expr1 이 NULL 이면 expr2, NULL 이 아니면 expr1 을 반환
SELECT IFNULL(a, 'UNKNOWN') FROM case_tbl;
```

- **인자 값 비교를 위한 IF(), NULLIF() 함수 추가 지원**

인자 값의 참/거짓 또는 인자 간 동등 비교를 수행하고, 결과에 따라 다른 값을 반환하는 함수를 추가 지원한다.

```
//expr1 이 참이면 expr2 를, 거짓이면 expr3 을 반환
SELECT IF(a=1, 'one', 'other') FROM case_tbl;

//expr1 과 expr2 가 동일하면 NULL 을 반환, 아니면 expr1 을 반환
SELECT NULLIF(a, 1) FROM case_tbl;
```

- 관련 문서: [조건 연산식과 함수> IFNULL 함수, NULLIF 함수, IF 함수](#) 등

새로 추가된 기능 - 기타

CUBRIDSUS-3005, 3085 데이터베이스 구동 중에도 공간 정리 작업을 수행할 수 있는 compactdb 유틸리티 옵션 지원

이전 버전에서는 데이터베이스 프로세스가 정지 상태인 경우에만 **cubrid compactdb** 유틸리티를 실행하여 공간 정리 작업을 수행할 수 있었으나, 구동 중인 상태(online)에서도 이를 수행할 수 있도록 클라이언트/서버 모드 실행 옵션(-C)을 추가 지원한다. 또한, 클라이언트/서버 모드에서만 적용할 수 있는 상세 옵션(-I, -i, -c, -d, -p)을 추가로 지원한다.

- 관련 문서: [데이터베이스 관리> 사용 공간 정리](#)

CUBRIDSUS-2923, 2961, 2972 PHP API에 20여 개의 함수 추가 지원

PHP API에 fetch 관련, field 정보 관련, DB 정보 관련 함수들을 추가로 지원한다.

Fetch 관련 함수	Field 관련 함수	DB 정보 관련 함수	기타 함수
cubrid_fetch_assoc()	cubrid_field_flags()	cubrid_get_client_info()	cubrid_data_seek()
cubrid_fetch_field()	cubrid_field_len()	cubrid_get_server_info()	cubrid_free_result()
cubrid_fetch_lengths()	cubrid_field_name()	cubrid_get_charset()	cubrid_insert_id()
cubrid_fetch_object()	cubrid_field_seek()	cubrid_get_db_parameter()	cubrid_result()
cubrid_fetch_row()	cubrid_field_table()	cubrid_list_dbs()	cubrid_unbuffered_query()
	cubrid_field_type()		cubrid_real_escape_string()
	cubrid_num_fields()		

- 관련 문서: [PHP API](#)

CUBRIDSUS-2829 insert_execution_mode 파라미터에서 설정할 수 있는 INSERT모드 추가

INSERT 문법 확장 및 **REPLACE** 문 지원으로 인해 **INSERT** 실행 모드가 추가되어, 총 5 개의 실행 모드의 조합으로 **insert_execution_mode** 파라미터 값을 설정할 수 있다. 이전 버전에서 지원된 **INSERT** 실행 모드는 INSERT_SELECT, INSERT_VALUES, INSERT_DEFAULT였으며, INSERT_REPLACE, INSERT_ON_DUP_KEY_UPDATE가 새로 추가되었다.

- 관련 문서: [데이터베이스 서버 설정> insert_execution_mode](#)

변경 및 개선된 사항

CUBRIDSUS-2855 cci_connect() 함수의 동작 방식 변경

CCI 라이브러리에서 데이터베이스 연결 관련 함수인 **cci_connect()** 및 **cci_connect_with_url()**의 동작 방식을 변경하였다. 이전 버전에서 해당 함수를 호출하면 주어진 연결 정보를 저장만 하고 실제로 연결 요청을 수행하지 않았으므로, 유효하지 않은 연결 정보가 주어지더라도 연결에 성공한 것으로 오인할 수 있었다. 이를 개선하여 해당 함수 호출 시 연결 정보에 대한 유효성을 검사하고, 실제 연결에 성공한 경우에만 연결 핸들을 반환하도록 수정하였다.

- 관련 문서: [CCI API> cci_connect, cci_connect_with_url](#)

CUBRIDSUS-3391 TO_CHAR() 함수에서 포맷 지정자 'HH', 'HH12'의 범위 변경

TO_CHAR 함수에서 'HH' 또는 'HH12' 지정자로 12시간 포맷을 사용하는 경우, 이전 버전에서는 시간 값이 0~11까지 출력되었으나, 이를 1~12까지 출력되도록 수정하였다. 이로 인해 업그레이드 이후, 응용 프로그램에서 해당 질의의 결과가 다를 수 있으므로 주의한다.

- 관련 문서: [데이터 타입 변환 함수와 연산자> TO_CHAR 함수](#)

CUBRIDSUS-3213 두 자리 수 연도 값에 대한 해석 범위 변경

이전 버전에서는 두 자리 수 연도 값 01~99을 0001~0099 년으로 해석하였으나, 이를 수정하여 00~69는 2000~2069 년으로, 70~99는 1970~1999 년으로 해석하도록 수정하였다. 이로 인해 업그레이드 이후, 응용 프로그램에서 해당 질의의 결과가 다를 수 있으므로 주의한다.

- 관련 문서: [데이터 타입 변환 함수> 날짜/시간 데이터 타입](#)

CUBRIDSUS-2703 POWER() 함수의 반환 타입을 DOUBLE로 변경

이전 버전에서는 **POWER()** 함수의 반환 타입이 주어진 인자에 의해 결정되었는데, 이를 인자 타입과 관계 없이 항상 DOUBLE 타입으로 반환하도록 수정하였다. 이로 인해 업그레이드 이후, 응용 프로그램에서 해당 질의의 결과가 다를 수 있으므로 주의한다. 또한, **POW()**를 동의어로 사용할 수 있다.

- 관련 문서: [수치 연산 함수> POWER 함수](#)

CUBRIDSUS-2421 조건절이 항상 FALSE인 특정 질의에서 순차 스캔이 발생되지 않도록 개선

GROUP BY 절이 있는 **SELECT** 문에서 집계 연산을 수행하는 질의에서, 조건절이 항상 **FALSE**인 경우에도 서버에서 순차 스캔을 수행하였으나, 이러한 알고리즘을 개선하여 조건절이 항상 **FALSE**인 경우 즉시 결과를 반환하도록 수정하였다.

CUBRIDSUS-961, 2890, 3030 JDBC 드라이버의 getPrimaryKeys() 및 getColumnDisplaySize() 메소드 지원

CUBRID JDBC 드라이버의 **CUBRIDDatabaseMetaData** 인터페이스에서 기본 키를 구성하는 컬럼 이름을 정렬하여 반환하는 **getPrimaryKeys()** 메소드를 지원한다. 그리고, 이전 버전에서는 **ResultSetMetaData** 인터페이스의 **getColumnDisplaySize()**가 precision 값이 없는 정수 타입 컬럼에 대해서는 0을 반환하고, VARCHAR 타입 컬럼에 대해서도 0을 반환하였으나, 실제 컬럼 사이즈를 반환하도록 이를 수정하였다.

CUBRIDSUS-2661 JDBC 드라이버에서 Spring 프레임워크의 DataAccessException 지원

Spring 프레임워크의 DataAccessException을 지원하여 보다 상세한 Exception 처리를 수행할 수 있도록 JDBC 드라이버를 개선하였다.

CUBRIDSUS-1737 Linux 환경에서 CUBRID 실행 스크립트 지원

Linux 환경에서 CUBRID가 자동 실행되도록 하는 **cubrid.init** 스크립트를 **CUBRID/contrib** 디렉토리에 제공한다. 이를 **/etc/init.d/**에 복사한 후 서비스에 등록할 수 있다.

CUBRIDSUS-1562, 1628 일부 테이블만 언로드 시, 지정된 입력 파일에 개행 문자가 없어도 정상 실행되도록 수정

데이터베이스의 일부 테이블만 언로드하기 위하여 **cubrid unloaddb** 유틸리티의 **-i** 옵션을 사용하는 경우, 이전 버전에서는 테이블 목록이 저장된 입력 파일의 마지막에 개행 문자(Wn)가 반드시 포함되어야 정상적으로 언로드 작업이 실행되었으나, 이를 수정하였다. 마지막에 개행 문자가 없거나, 라인 중간에 구분자로 개행 문자를 사용한 경우에도 정상 실행된다.

이와 관련하여, Windows 환경에서 환경 설정 파일(*.conf)의 마지막 라인에서 파라미터 입력 후 엔터 키를 입력하지 않을 경우, 이전 버전에서는 해당 파라미터가 적용되지 않았으나 이를 수정하였다.

CUBRIDSUS-1216 HA 환경에서 마스터 서버의 장애 복구 이후, RW모드의 브로커를 자동 초기화하는 기능 지원

HA 환경에서 마스터 서버의 장애가 복구된 이후, 이전 버전에서는 쓰기 작업이 요청되거나 브로커 연결을 초기화하는 명령이 입력되기 전까지는 일부 CAS 프로세스가 슬레이브 서버와의 연결을 유지하도록 동작하였다. 이를 개선하여, 마스터 서버 프로세스가 재시작되면 RW 모드의 브로커가 자동으로 연결 상태를 초기화하여 슬레이브 서버와 연결되어 있는 CAS 프로세스가 처리 중인 트랜잭션 종료 이후 마스터 서버로 연결되도록 수정하였다.

CUBRIDSUS-1970 계층 질의문에서 인덱스 스캔을 수행하도록 수정

이전 버전에서는 계층 질의문의 대상 테이블에 인덱스가 정의되어 있더라도 순차 스캔만 수행하였으나, 인덱스 스캔을 수행하도록 개선하였다.

수정된 오류

CUBRIDSUS-3194,3399,3434,3435 HA 환경에서 applylogdb 프로세스의 CPU 사용률이 증가하는 오류 수정

HA 환경에서 복제 로그를 슬레이브 DB로 반영할 때 NULL 페이지를 처리하는 알고리즘의 오류로 인해 applylogdb 프로세스의 CPU 사용률이 지속적으로 증가하는 현상이 발생하였으나, 이를 수정하였다. 또한, applylogdb 프로세스가 특정 구간의 로그를 반복적으로 반영하는 오류로 인해 CPU 사용률이 증가하면서 복제가 중단되는 현상이 발생하였으나, 이 오류도 함께 수정하였다.

CUBRIDSUS-3196, 3489 HA 환경에서 applylogdb 프로세스가 반복적으로 재시작되며 복제 진행되지 않는 현상 수정

HA 환경에서 마스터 DB의 복제 로그 정보와 슬레이브 DB의 applylogdb 프로세스가 반영하려는 페이지 정보의 불일치로 인해 applylogdb 프로세스가 반복적으로 재시작되면서 복제가 진행되지 않는 오류가 있었으나, 이를 수정하였다. 이러한 오류는 쓰기 연산이 드물게 수행됨에도 불구하고 데이터베이스를 자주 백업하는 환경에서 발생할 수 있다.

CUBRIDSUS-3106 HA 환경에서 특정 커밋 로그 유실로 인해 복제가 중단되는 오류 수정

HA 환경에서 특정 아카이브 로그에 의도하지 않은 EOL(End-Of-Log)가 기록되는 오류로 인해 트랜잭션 커밋 로그가 유실되어 복제가 중단되는 현상이 발생하였으나, 이를 수정하였다.

CUBRIDSUS-2831 HA 환경 구성 직후 클라이언트 모드에서 backupdb 유틸리티 수행 시 작업이 중지되는 오류 수정

HA 환경 구성 후 새로 생성한 DB를 백업하기 위하여 **cubrid backupdb** 유틸리티를 클라이언트 모드(-C 옵션)에서 수행하는 경우, 백업 작업이 중지되는 오류가 있었으나 이를 수정하였다.

CUBRIDSUS-2872 HA 환경에서 CUBRID heartbeat 노드 상태 정보 불일치 오류 수정

CUBRID heartbeat을 이용하여 HA를 구성하는 경우, heartbeat 메시지 수신 알고리즘 오류로 인해 HA 구성된 노드가 전부 master로 인식되는 문제가 있었으나, 이를 수정하였다.

- 발생 버전: CUBRID 2008 R2.2

CUBRIDSUS-2120 복제/HA 환경에서 외래 키가 정의된 테이블에 INSERT 수행 시 데이터 불일치 오류 수정

복제 또는 HA 환경에서 기본 키 테이블 및 외래 키 참조 제약 조건이 정의된 테이블이 존재하고, 트랜잭션 A 및 트랜잭션 B가 각각 기본 키 테이블 및 외래 키 테이블에 **INSERT**를 수행하는 경우, 트랜잭션 A와 B의 커밋 순서에 따라서 데이터 불일치가 발생할 수 있었으나, 이러한 불일치가 발생하지 않도록 수정하였다.

CUBRIDSUS-3295, 3191 복제 환경에서 데이터 불일치 오류 수정

마스터 DB에서 트랜잭션 도중 CAS 프로세스에서 작업 취소(query_cancel)을 수행하고 해당 연산을 롤백 처리한 경우, 롤백한 연산에 대한 복제 로그가 생성되는 오류로 인해 마스터 DB 및 슬레이브 DB의 데이터 불일치 현상이 발생하였으나, 이를 수정하였다.

CUBRIDSUS-3135 복제 환경에서 복제 로그 크기가 비정상적으로 증가하는 오류 수정

마스터 DB에서 복제가 지연되는 특정 상황에서 복제 로그가 슬레이브 DB로 복사된 이후에도 마스터 DB의 복제 로그 크기가 비정상적으로 증가하는 오류가 있었으나, 이를 수정하였다.

CUBRIDSUS-2638 특정 상황에서 데이터베이스 복구 작업 수행 후 질의가

정상 수행되지 않는 오류 수정

백업이 수행된 이후 트랜잭션 A에 의해 새로운 볼륨이 추가된 이후 커밋 전에 서버 프로세스가 종료되고, 볼륨 추가에 관한 로그가 디스크에 출력되는 경우, 볼륨 정보(volume info)와 실제 존재하는 볼륨 파일이 일치하지 않는 상태가 된다. 이런 상태에서 대상 데이터베이스에 대한 복구(restore) 작업이 수행되면, 이전 버전에서는 추가된 볼륨을 누락한 채 복구 작업을 완료하여 실제로 대상 데이터베이스에 질의 수행 시 Internal Error가 출력되는 오류가 있었으나, 이를 수정하였다.

CUBRIDSUS-3083 UNIQUE KEY 컬럼에 중복 값이 삽입되는 오류 수정

단일 **UPDATE** 문을 사용하여 다중 행의 컬럼 값을 변경하는 경우, **UNIQUE** 제약 조건이 정의된 컬럼에 중복된 값이 저장되는 오류를 수정하였다.

- 발생 버전: CUBRID 2008 R2.2

CUBRIDSUS-2127 조인 조건식 내에서 특정 인덱스 키 값에 대해 OR

연산이 적용되지 않는 오류 수정

인덱스가 설정된 두 개 이상의 테이블이 조인되는 질의문에서 **OR**로 연결된 왼쪽 조건식의 결과가 없는(empty result) 경우에 오른쪽 조건식을 판단하지 않는 오류로 인해 잘못된 질의 결과를 반환하는 문제가 있었으나, 이를 수정하였다.

CUBRIDSUS-2062 외부 조인 수행 시, 질의 결과 오류 수정

외부 조인(outer join) 질의문의 동작 오류로 인해 이전 버전에서는 왼쪽 외부 조인 질의문과 오른쪽 외부 조인 질의문을 실행하면 각각 다른 결과가 출력되었으나, 이를 수정하였다.

CUBRIDSUS-423 inline-view 형식의 질의문의 결과에서 컬럼 이름이

출력되지 않는 오류 수정

아래와 같이 inline-view 형식의 질의문으로 테이블의 oid 값을 조회하는 경우, 컬럼에 저장된 oid 값은 정상 출력되나 컬럼 이름이 공백으로 출력되는 오류가 있었으나, 이를 수정하였다.

```
SELECT * FROM (SELECT tbl_1 FROM tbl_1) A;
```

CUBRIDSUS-3337 SELECT 리스트에 ORDER BY 절이 존재하는 상관

질의가 명시되는 경우, 질의 결과 오류 수정

ORDER BY 절을 가지는 상관 질의(correlated sub-query)가 두 개 이상 **SELECT** 리스트에 명시되는 경우, 해당 부 질의를 내부적으로 재작성(rewrite)하는 알고리즘의 오류로 인해 잘못된 질의 결과가 출력되는 오류가 있었으나, 이를 수정하였다.

```
//오류가 재현된 질의 유형
SELECT
  (SELECT nvl(id2,id1) FROM t1 WHERE id = t o.id ORDER BY id2, id1 FOR ORDERBY NUM() = 1) AS col1,
  (SELECT count(val) FROM t2 WHERE id = t o.id and val> 0) AS col2
FROM tbl_out t_o WHERE rownum < 3;
```

CUBRIDSUS-2742 FOR ORDERBY_NUM()가 있는 부 질의가 포함된 경우,

질의 결과 오류 수정

FOR ORDERBY_NUM() 절을 가지는 부 질의가 **SELECT** 리스트에 명시되는 경우, 해당 부 질의를 내부적으로 재작성(rewrite)하는 알고리즘의 오류로 인해 잘못된 질의 결과가 출력되는 오류가 있었으나, 이를 수정하였다.

```
//오류가 재현된 질의 유형
SELECT
(SELECT a FROM xoo ORDER BY b ASC, a ASC FOR ORDERBY_NUM() = 1 ) c1,
b
FROM xoo x ORDER BY order_no ASC;
```

CUBRIDSUS-2530 특정 계층 질의문에서 레벨 값 출력 오류 수정

레코드가 삭제됨에 따라 해당 레코드의 저장 공간에 새로운 레코드가 저장되는 경우, 계층 질의문 실행 결과 잘못된 LEVEL 값이 출력되는 오류가 있었으나, 이를 수정하였다. 단, 이 오류는 공간 재사용으로 인해 발생되므로 **dont_reuse_heap** 파라미터 값을 yes로 설정하면 발생하지 않는다.

CUBRIDSUS-2149 인덱스 컬럼에 대해 ON CACHE OBJECT 속성의 외래 키를 정의하는 경우, 참조 값이 업데이트되지 않는 오류 수정

이미 인덱스가 정의된 컬럼에 대해 **ON CACHE OBJECT** 속성의 외래 키를 생성하는 경우, 해당 컬럼이 참조하는 값이 업데이트되지 않는 오류를 수정하였다.

CUBRIDSUS-2603 CREATE SERIAL, ALTER SERIAL 문의 START WITH 옵션 오류 수정

CREATE SERIAL 문 또는 **ALTER SERIAL** 문에서 **START WITH** 옵션이 첫 번째로 명시되는 경우에만 정상 동작하고 중간에 명시되는 경우 오류를 출력하였으나, 이를 옵션이 명시되는 순서에 무관하게 정상 동작하도록 수정하였다.

CUBRIDSUS-3003 ORDER BY 절을 포함한 부질의 결과를 SEQUENCE로 표현할 때 부질의 결과가 정렬되지 않는 오류 수정

ORDER BY 절을 포함한 부질의 결과를 **SEQUENCE** 또는 **LIST** 타입으로 표현하려고 할 때 부질의 결과가 **ORDER BY** 절에 의해 정렬되지 않는 오류가 있었으나 이를 수정하였다.

CUBRIDSUS-3299 CREATE VIEW 문에서 컬럼 이름을 명시하지 않는 경우, VIEW의 컬럼 타입이 잘못 생성되는 오류 수정

VIEW 생성 시 대상 **SELECT** 절에서 컬럼 이름을 명시하지 않는 경우, 자동으로 생성되는 컬럼 타입의 오류를 수정하였다.

```
//이전 버전은 VARCHAR(1073741823) 타입으로 생성했으나, 새 버전에서는 VARCHAR(3)으로 생성됨
CREATE VIEW vx AS SELECT 'aaa' a FROM db root;

//이전 버전은 컬럼 타입이 none 이었으나, 새 버전에서는 INTEGER 로 생성됨
CREATE VIEW AS SELECT 1 + 1 a FROM db_root;
```

CUBRIDSUS-3203, 3224, 3246 CAST 함수를 사용한 명시적 타입 변환 수행 시 오류 수정

NUMERIC 타입 값을 **BIGINT** 타입으로 명시적 타입 변환(coerce)하는 경우, .5가 반올림 처리되지 않는 오류를 수정하였다. **DATETIME** 타입을 문자열 타입으로 변환하는 경우, 문자열 마지막에 공백이 삽입되는 오류를 수정하였다.

문자열을 숫자 타입으로 변환하는 경우, 문자열의 앞 또는 뒤에 공백이 있는 경우 잘못된 값으로 변환되는 오류를 수정하였다.

CUBRIDSUS-3491 멀티쓰레드 환경에서 REUSE_OID 테이블에

INSERT/DELETE 연산을 반복 수행 시, 서버 오류 출력되는 문제 수정

멀티쓰레드 환경에서 **REUSE_OID** 옵션이 적용된 테이블에 대해 **INSERT** 연산과 **DELETE** 연산을 반복적으로 수행하는 경우, 아래와 같은 서버 오류가 발생하는 문제를 수정하였다. 이러한 문제는 5개의 쓰레드가 **INSERT** 연산과 **DELETE** 연산을 200개씩 수행하는 환경에서 재현되었다.

```
Time: 06/30/10 16:14:08.309 - ERROR *** ERROR CODE = -45, Tran = 3, CLIENT =
cdbs005.cub:broker1_cub_cas_3(10674), EID = 1
Slot 2 on page 50 of volume "/DB/foo_x001" is allocated to an anchored record. A new record cannot
be inserted here.
```

CUBRIDSUS-440 트리거가 정의된 클래스를 삭제한 경우, 시스템

카탈로그 뷰에서 트리거 정보가 검색되지 않는 오류 수정

트리거가 정의된 클래스를 삭제하였으나 해당 트리거를 수동 삭제하지 않은 경우, 이전 버전에서는 시스템 카탈로그 테이블인 **db_trigger**에서 해당 트리거가 검색되나, 시스템 카탈로그 뷰인 **db_trig**에서는 해당 트리거가 검색되지 않는 오류가 있었다. 이를 수정하여 시스템 카탈로그 뷰에서도 삭제되지 않은 트리거 정보를 검색할 수 있도록 하였다.

CUBRIDSUS-1559 COMMIT 또는 ROLLBACK 이벤트 타입 트리거가

언로드되지 않는 오류 수정

이벤트 대상 테이블이 존재하지 않는 COMMIT 또는 ROLLBACK 이벤트 타입의 트리거가 정의된 경우, 해당 트리거가 누락된 채 DB 언로드 작업이 수행되는 오류가 있었으나 이를 수정하였다.

CUBRIDSUS-2774 broker_log_top 유틸리티의 동작 오류 수정 및 브로커

로그 포맷 변경

응용 프로그램에서 BIT 타입 값이 바인딩되는 경우, **broker_log_top**, **broker_log_converter**, **broker_log_runner** 유틸리티가 동작하지 않는 오류를 수정하고 브로커 로그 포맷을 변경하였다. 이로 인해, CUBRID 2008 R3.0 버전에서 생성된 브로커 로그는 하위 버전의 브로커 로그와 호환되지 않으므로 R1.x, R2.x 버전 브로커 로그를 대상으로 위의 브로커 로그 관련 유틸리티를 실행할 수 없다. 마찬가지로, R3.0 버전 브로커 로그를 대상으로 R1.x, R2.x 버전의 브로커 로그 관련 유틸리티를 실행할 수 없다.

CUBRIDSUS-2886 JDBC 를 통한 서버 접속 요청이 집중되는 환경에서

접속 실패하는 현상 수정

JDBC 드라이버를 통해 수십 개의 쓰레드가 집중적으로 서버에 접속하여 **INSERT**를 수행하는 환경에서 아래의 에러 메시지가 출력되면서 접속 실패하는 현상이 발생하였으나, 이를 수정하였다.

- JDBC 에러 메시지

```
cubrid.jdbc.driver.CUBRIDException: Failed to connect to database server, 'address', on the
following host(s): localhost
at cubrid.jdbc.driver.CUBRIDConnection.prepare(CUBRIDConnection.java:676)
at cubrid.jdbc.driver.CUBRIDConnection.prepare(CUBRIDConnection.java:917)
at cubrid.jdbc.driver.CUBRIDConnection.prepareStatement(CUBRIDConnection.java:137)
at worker.addr insert(foo address.java:170)
at worker.run(foo_address.java:44)
```

- 브로커 에러 로그

```
Time: 04/29/10 19:36:10.258 - ERROR *** ERROR CODE = -353, Tran = -1, EID = 1
Cannot make connection to master server.... Resource temporarily unavailable

Time: 04/29/10 19:36:10.258 - ERROR *** ERROR CODE = -677, Tran = -1, EID = 2
Failed to connect to database server, 'address', on the following host(s): localhost
```

- 발생 버전: CUBRID 2008 R2.0 Patch2, R2.2

CUBRIDSUS-3367, 3427 Java Stored Procedure에서 ResultSet 반환 오류 및 cci_cursor_update() 동작 오류 수정

Java Stored Procedure를 사용하는 JDBC 응용에서 JSP를 통해서 반환된 resultset을 접근할 때 `Missing or invalid position of the bind variable provided.`라는 메시지가 출력되면서 정상 동작하지 않는 문제가 있었다. 이와 동일한 오류로 인해 CCI 인터페이스에서도 `cci_cursor_update()` 함수가 정상 동작하지 않는 문제가 있었으나 이를 수정하였다.

CUBRIDSUS-3137 DATETIME 타입 값에 대해 JDBC 드라이버의 getString() 출력 값 오류 수정

JDBC 드라이버에서 `getString()` 메소드를 호출하여 DATETIME 타입 값을 출력하는 경우, 이전 버전에서는 밀리 초 단위의 값이 출력되지 않는 오류가 있었으나 이를 수정하였다.

CUBRIDSUS-3457 컬렉션 타입 컬럼에 대한 cci_col_get() 함수의 동작 오류 수정

컬렉션 타입 컬럼에 대해 `cci_col_get()` 함수를 사용하는 경우 응용 프로그램이 비정상 종료되거나 잘못된 결과를 반환하는 오류를 수정하였다.

CUBRIDSUS-3523 CCI 및 PHP 응용 환경에서 cci.log 파일이 생성되는 문제 수정

CCI 라이브러리의 오류로 인해 CUBRID 실행 디렉토리 내에 cci.log 라는 불필요한 로그 파일이 생성되는 문제를 수정하였다.

CUBRIDSUS-2225 databases.txt에서 구분자가 tab인 경우에도 cubrid backupdb -sp 옵션이 정상 동작하도록 수정

복제 환경에서 `cubrid backupdb` 유틸리티를 수행하는 경우, 백업 후 보관 로그의 안전한 삭제를 위해 `-r` 옵션과 `-sp` 옵션을 함께 명시한다. `databases.txt` 파일에 포함된 DB 명 및 DB 경로가 공백이 아닌 탭(tab)으로 구분된 경우, 이전 버전에서는 `-sp` 옵션의 인자로 지정되는 복구 스크립트(repl_safe_page)가 파일로부터 DB 경로 정보를 가져오지 못하여 옵션이 적용되지 않은 채 백업이 수행되었으나, 이를 수정하여 구분자가 탭인 경우에도 해당 옵션이 정상 적용되도록 하였다.

CUBRIDSUS-2221 cub_master 프로세스가 허용하는 연결 시도 요청만 처리하도록 수정

`cub_master` 프로세스가 사용 중인 포트로 외부 프로세스가 연결을 요청하는 경우, 이전 버전에서는 유효하지 않은 연결 요청을 감지하지 못하는 오류가 있었다. 이로 인한 안정성 문제에 대비하고자 `cub_master` 프로세스의 프로토콜을 개선하여 유효하지 않은 연결 시도 요청을 즉시 종료하도록 수정하였다.

CUBRIDSUS-1199 브로커 상태 모니터링 수행 시, 특정 항목 값의 출력

오류 수정

`cubrid broker status -s 5 <broker_name>` 명령어를 실행하여 주기적으로 브로커 상태 정보를 모니터링하는 도중, 화살표 키가 입력되면 QPS(Query per Second), LQS(Long Query per Second) 항목 값이 음수로 출력되는 오류를 수정하였다.

CUBRIDSUS-2019 CAS 프로세스의 메모리 크기가 2G 초과 시,

재시작되지 않는 오류 수정

CAS 프로세스의 메모리 크기가 2G 를 초과하는 특정 상황에서 브로커 모니터링 항목 중 하나인 PSIZE 값이 음수로 출력되고, 이로 인해 해당 프로세스가 재시작되지 않는 오류가 있었으나 이를 수정하였다.

CUBRIDSUS-2047 media_failure_support = no로 설정된 경우, 임시 보관

로그 크기가 무한 증가하는 오류 수정

`background_archiving` 파라미터는 `media_failure_support = yes` 인 경우에만 주기적으로 임시 보관 로그를 생성하도록 설정하는 파라미터인데, 이전 버전에서는 `media_failure_support = no` 인 경우에도 `background_archiving` 파라미터가 적용되는 오류로 인해 임시 보관 로그 파일(lgar_t)의 크기가 무한 증가하는 현상이 발생하였으나, 이를 수정하였다.

CUBRIDSUS-343 CSQL에서 50 라인 이상 질의문에 대한 이전 질의문

불러오기 기능의 오류 수정

CSQL 인터프리터에서 50 라인 이상의 질의를 수행한 후 화살표 Up 버튼을 입력하여 이전 질의문 불러오기 기능을 실행하면 서버 프로세스가 중단되는 오류가 있었으나, 이를 수정하였다.

CUBRIDSUS-600 Windows Vista 환경에서 CSQL 세션 명령어 ;edit 실행

시 발생하는 오류 수정

Windows Vista 환경에서 CSQL 세션 명령어 `;edit` 실행 시 `ERROR: No such file or directory` 라는 메시지가 출력되는 오류를 수정하였다.

CUBRIDSUS-3592, 3359 Windows 환경에서 CURBID 설치 오류 수정

Windows 환경에서 CUBRID 설치 도중 Microsoft Visual C++ 2008 재배포 가능 패키지 설치를 요청하는 알림 창이 잘못 출력되는 오류를 수정하고, 설치 도중 `ctrlService.exe`의 작동이 중지되었습니다. 라는 알림 창이 출력되면서 설치가 진행되지 않는 오류를 수정하였다.

CUBRIDSUS-471 Windows 환경에서 cubrid service stop/start 명령의

실행 결과가 CUBRID 트레이에 반영되지 않는 오류 수정

Windows 환경에서 실행 창에 `cubrid service start/stop/restart`와 같은 CUBRID 서비스 명령을 실행하는 경우, 이전 버전에서는 이러한 실행 명령이 윈도우 서비스에 반영되지 않아 CUBRID 트레이 상태가 변경되지 않는 오류가 있었다. 이를 수정하여 Windows 실행 창에서 입력된 CUBRID 서비스 명령이 윈도우 서비스에 정상 반영되도록 하였다.

4. CUBRID 매니저 2008 R3.0 버전에서 변경된 사항

새로 추가된 기능

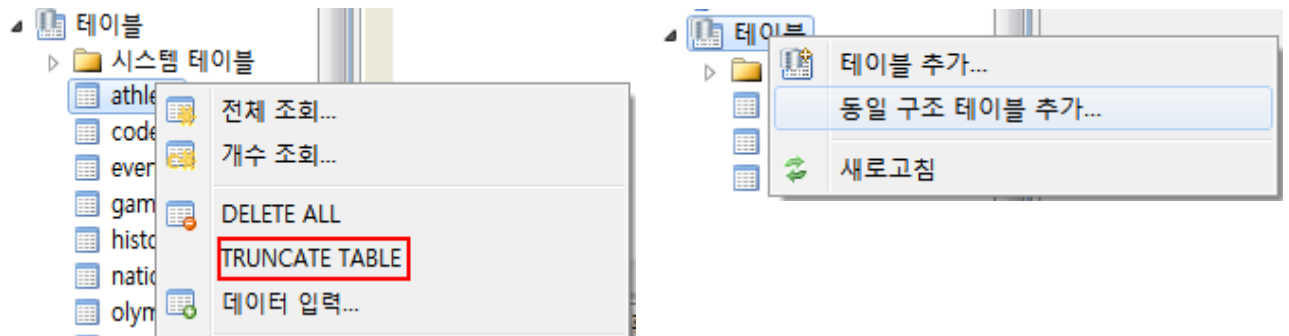
HA 기능 확대 지원

CUBRID 매니저를 사용하여 HA 환경으로 구성된 슬레이브 DB에 DML 작업을 수행할 수 있고, 슬레이브 DB를 백업할 수 있다. 또한, 이전 버전에서는 마스터 서버 장애로 인해 절체(failover)가 수행되면 CUBRID 매니저가 슬레이브 서버로 접속을 변경하여 사용자가 의도하지 않은 DB작업이 슬레이브 서버에 반영되는 오류가 있었으나, 이를 수정하였다.

추가된 구문을 위한 신규 키워드 및 UI 메뉴 지원

CUBRID 매니저의 질의 편집기에서 이번 버전에서 추가된 새로운 연산자, 함수 및 확장된 구문을 실행할 수 있도록 키워드 및 UI 메뉴를 추가 지원한다. 단, UI 메뉴는 스키마 변경과 관련된 특정 기능(**TRUNCATE TABLE**, **CREATE TABLE LIKE**)에 대해서만 지원되며, 이 메뉴는 CUBRID 2008 R3.0 이상 버전의 서버에 접속하는 경우에만 활성화된다.

- 테이블 팝업 메뉴로 [TRUNCATE TABLE] 및 [동일 구조 테이블 추가] 기능 추가



- 테이블 생성/편집 창에서 외래 키 옵션으로 SET NULL을 설정할 수 있도록 개선

테이블 생성/편집 창에서 외래 키를 설정하는 경우, 기본 키에 대한 업데이트 또는 삭제 연산에 대한 트리거 동작으로 SET NULL 옵션을 설정할 수 있다.

- 관련 문서: [CUBRID 매니저 > 데이터베이스 > 테이블](#)

[상태 모니터 추가]에서 사용자 설정 기능 지원

[상태 모니터 > 상태 모니터 추가] 메뉴에서 차트 제목, 그림 모양, 모니터링 항목 등을 사용자가 설정할 수 있는 기능을 지원하여, 사용자가 원하는 차트 템플릿을 생성할 수 있다.

- 관련 문서: [CUBRID 매니저 > 상태 모니터](#)

변경 및 개선된 사항

[새 질의 편집기] 실행 시 원격의 브로커 호스트 정보를 설정할 수 있는 기능 지원

[도구 > 새 질의 편집기] 또는 [툴 바 > 새 질의 편집기 아이콘]을 선택하면 데이터베이스 로그인 창이 출력되는데, 이 창에서 데이터베이스 정보 및 브로커 정보를 각각 설정할 수 있는 기능을 지원하여 보다 편리하게 질의 편집기 실행 정보를 입력할 수 있도록 하였다. 단, 브로커 포트는 매니저 사용자 계정이 admin 인 경우에만 변경 가능하며,

기본으로 출력되는 문자 집합은 질의를 실행하고자 하는 대상 데이터베이스의 문자 집합이 출력된다. 만약, 사용자가 문자 집합 항목을 설정하지 않으면 현재 CUBRID 매니저가 구동 중인 시스템(PC)의 문자 집합이 설정된다.

[질의 자동화]에 사용자 로그인 기능 지원

[작업 자동화 > 질의 자동화]를 사용하여 특정 질의를 자동 수행하는 경우, 해당 질의를 수행할 데이터베이스 사용자 계정 및 비밀번호를 입력 받아 해당 질의가 수행되는 시점에 실행 권한을 확인하도록 개선하였다. 잘못된 사용자 정보가 입력된 경우에는 해당 질의가 자동 수행되지 않는다.

cm.conf의 파라미터 값 설정 방식 확장

이전 버전에서는 **cm.conf** 내 파라미터 설정 시 파라미터 이름과 값 사이의 구분자로 공백(space)만 사용할 수 있었으나, 등호(=)를 사용할 수 있도록 개선하였다.

특정 경로에 데이터베이스를 복구할 수 있는 기능 지원

[데이터베이스 복구] 창에서 복구(restore) 작업을 수행하는 경우, 대상 데이터베이스가 복구되는 위치를 사용자가 설정할 수 있는 [복구 경로] 옵션을 추가 지원한다.

수정된 오류

[데이터베이스 언로드] 기능 수행 시, 테이블 일부 언로드가 수행되지 않는 오류 수정

[데이터베이스 언로드] 기능을 사용하여 사용자가 선택한 일부 테이블에 대해서만 부분 언로드를 수행하는 경우에도 전체 언로드가 수행되는 오류가 있었으나, 이를 수정하였다.


다중 질의 실행 시 일부 질의 오류에 대한 경고 창이 계속 출력되는 오류 수정

질의 편집기에서 여러 개의 질의문을 한번에 실행하고 그 중 일부 질의문에 오류가 발견되는 경우, 나머지 질의문을 계속 실행할 것인지를 확인하는 경고 창이 출력된다. 이전 버전에서는 사용자가 [다음에 묻지 않음] 옵션을 선택하더라도 이후 발견되는 질의 오류에 대해 경고 창이 반복적으로 출력되는 오류가 있었으나 이를 수정하였다. 따라서, 오류가 발견된 질의문을 제외한 정상 질의문에 대해서만 질의 결과가 출력된다.

질의 편집기에서 조회 결과가 5000건을 초과하는 경우 조회를 중단할 수 없는 오류 수정

질의 편집기는 5,000건 단위로 레코드를 조회하는데, 조회 결과가 이를 초과하는 경우 계속 조회 진행 또는 조회 중단을 선택할 수 있는 경고 창이 출력된다. 이전 버전에서는 [아니오]를 선택하더라도 이후 결과 레코드 전부를 출력하는 오류가 있었으나 이를 수정하였다.

특정 상황에서 매니저가 종료되지 않는 오류 수정

CUBRID 매니저를 장시간 구동하는 경우, 매니저 종료 버튼()이 동작하지 않는 오류가 있었으나 이를 수정하였다.

JDK 1.5 버전 환경에서 질의 편집기에서 수행된 질의 결과가 비정상 출력되는 오류 수정

JDK 1.5 환경에서 매니저의 질의 편집기를 사용하여 질의를 수행하는 경우, `java/sql/NClob not found` 메시지가 출력되는 오류를 수정하였다.

5. 주의 사항

CUBRIDSUS-3217 JDBC에서 연결 정보를 URL 스트링으로 입력하는 경우

물음표를 반드시 명시

JDBC에서 URL 스트링으로 연결 정보를 입력하는 경우, 이전 버전에서는 물음표(?)를 입력하지 않더라도 속성(Property) 정보가 적용되었으나, 이번 버전부터는 문법에 따라 반드시 물음표를 명시하여야 하고 이를 생략할 경우 에러를 출력한다. 또한, 연결 정보 중 USERNAME과 PASSWORD가 없더라도 반드시 콜론(:)을 명시하여야 한다.

```
URL=jdbc:CUBRID:127.0.0.1:31000:db1::althosts=127.0.0.2:31000,127.0.0.3:31000 -에러 처리
URL=jdbc:CUBRID:127.0.0.1:31000:db1::?althosts=127.0.0.2:31000,127.0.0.3:31000 -정상 처리
```

CUBRIDSUS-3564 마스터와 서버 프로세스 간 프로토콜 변경 및 두 개

버전을 동시에 운영하는 경우 포트 설정 필요

마스터 프로세스와 서버 프로세스 간 통신 프로토콜 변경으로 인해 CUBRID 2008 R3.0 버전의 마스터 프로세스는 하위 버전의 서버 프로세스와 통신할 수 없고, 하위 버전의 마스터 프로세스도 R3.0 버전의 서버 프로세스와 통신할 수 없다. 따라서, 이미 하위 버전이 설치되어 있는 환경에서 새 버전을 추가 설치하여, 두 개 버전의 CUBRID를 동시에 운영하는 경우, 각각 서로 다른 포트를 사용하도록 **cubrid.conf**의 **cubrid_port_id** 파라미터를 수정하여야 한다.

CUBRIDSUS-2828 데이터베이스 이름에 @를 포함할 수 없음

데이터베이스 이름에 @이 포함되는 경우 호스트 이름이 명시된 것으로 해석될 수 있으므로, 이를 방지하기 위하여 **cubrid createdb**, **cubrid renamedb**, **cubrid copydb** 유틸리티 실행 시 데이터베이스 이름에 @를 포함할 수 없도록 수정하였다.

CUBRIDSUS-3267 Windows 환경에서 디렉토리 경로 설정 시 주의 사항

Windows 환경에서 CUBRID 설치 디렉토리 경로에 공백을 포함하는 경우 정상 설치가 되지 않으므로 주의한다. 또한, DB 언로드/로드/백업 등의 작업 대상 디렉토리 경로에도 공백을 포함할 수 없다.

CUBRIDSUS-3553 CUBRID 소스 빌드 후 실행 시, 매니저 서버 프로세스

관련 오류 발생

사용자가 직접 빌드하여 설치하는 경우, CUBRID와 CUBRID Manager를 각각 빌드하여 설치하여야 한다. 만약, CUBRID 소스만 checkout하여 빌드 후 **cubrid service** 또는 **cubrid manager**를 실행하면, **cubrid manager server is not installed**라는 오류가 발생한다.

CUBRID 매니저에서 [볼륨 자동 추가 기능 사용] 옵션을 선택하고 DB를

생성하는 경우 주의 사항

[볼륨 자동 추가 기능 사용] 옵션은 DB 볼륨이 사용자가 설정한 "여유 공간 비율" 값 이하가 될 경우 해당 볼륨을 자동으로 추가하는 기능이며, 이 옵션이 선택되면 여유 공간을 모니터링하기 위해 주기적(디폴트 5초)으로 **cubrid spacedb** 유틸리티를 수행하므로 에러 로그(<dbname>_spacedb.err)가 증가할 수 있다. 사용자는 **cm.conf** 파일의 **monitor_interval** 파라미터를 설정하여 모니터링 주기를 조정할 수 있다.