

Tibero

tbCLI 안내서

Tibero 6



Copyright © 2020 TmaxData Co., Ltd. All Rights Reserved.

Copyright Notice

Copyright © 2020 TmaxData Co., Ltd. All Rights Reserved.

대한민국 경기도 성남시 분당구 황새울로258번길 29, BS 타워 9층 우)13595

Website

<http://www.tmaxdata.com>

기술서비스센터

Tel : +82-1544-8629

E-Mail : info@tmax.co.kr

Restricted Rights Legend

All TmaxData Software (Tibero®) and documents are protected by copyright laws and international convention. TmaxData software and documents are made available under the terms of the TmaxData License Agreement and this document may only be distributed or copied in accordance with the terms of this agreement. No part of this document may be transmitted, copied, deployed, or reproduced in any form or by any means, electronic, mechanical, or optical, without the prior written consent of TmaxData Co., Ltd. Nothing in this software document and agreement constitutes a transfer of intellectual property rights regardless of whether or not such rights are registered) or any rights to TmaxData trademarks, logos, or any other brand features.

This document is for information purposes only. The company assumes no direct or indirect responsibilities for the contents of this document, and does not guarantee that the information contained in this document satisfies certain legal or commercial conditions. The information contained in this document is subject to change without prior notice due to product upgrades or updates. The company assumes no liability for any errors in this document.

이 소프트웨어(Tibero®) 사용설명서의 내용과 프로그램은 저작권법과 국제 조약에 의해서 보호받고 있습니다. 사용설명서의 내용과 여기에 설명된 프로그램은 TmaxData Co., Ltd.와의 사용권 계약 하에서만 사용이 가능하며, 사용설명서는 사용권 계약의 범위 내에서만 배포 또는 복제할 수 있습니다. 이 사용설명서의 전부 또는 일부분을 TmaxData의 사전 서면 동의 없이 전자, 기계, 녹음 등의 수단을 사용하여 전송, 복제, 배포, 2차적 저작물작성 등의 행위를 하여서는 안 됩니다.

이 소프트웨어 사용설명서와 프로그램의 사용권 계약은 어떠한 경우에도 사용설명서 및 프로그램과 관련된 지적 재산권(등록 여부를 불문)을 양도하는 것으로 해석되지 아니하며, 브랜드나 로고, 상표 등을 사용할 권한을 부여하지 않습니다. 사용설명서는 오로지 정보의 제공만을 목적으로 하고, 이로 인한 계약상의 직접적 또는 간접적 책임을 지지 아니하며, 사용설명서 상의 내용은 법적 또는 상업적인 특정한 조건을 만족시키는 것을 보장하지는 않습니다. 사용설명서의 내용은 제품의 업그레이드나 수정에 따라 그 내용이 예고 없이 변경될 수 있으며, 내용상의 오류가 없음을 보장하지 아니합니다.

Trademarks

Tibero® is a registered trademark of TmaxData Co., Ltd. Other products, titles or services may be registered trademarks of their respective companies.

Tibero®는 TmaxData Co., Ltd.의 등록 상표입니다. 기타 모든 제품들과 회사 이름은 각각 해당 소유주의 상표로서 참조용으로만 사용됩니다.

Open Source Software Notice

Some modules or files of this product are subject to the terms of the following licenses. : OpenSSL, RSA Data Security, Inc., Apache Foundation, Jean-loup Gailly and Mark Adler, Paul Hsieh's hash

Detailed Information related to the license can be found in the following directory : \${INSTALL_PATH}/license/oss_licenses

본 제품의 일부 파일 또는 모듈은 다음의 라이선스를 준수합니다. : OpenSSL, RSA Data Security, Inc., Apache Foundation, Jean-loup Gailly and Mark Adler, Paul Hsieh's hash

관련 상세한 정보는 제품의 다음의 디렉터리에 기재된 사항을 참고해 주십시오. : \${INSTALL_PATH}/license/oss_licenses

안내서 정보

안내서 제목: Tibero tbCLI 안내서

발행일: 2024-03-29

소프트웨어 버전: Tibero 6.7.1

안내서 버전: v6.7.1

내용 목차

안내서에 대하여	xiii
제1장 tbCLI 소개	1
1.1. 개요	1
1.2. 구성요소	2
1.2.1. tbCLI 핸들	2
1.2.2. tbCLI 함수	3
1.2.3. tbCLI 에러 메시지	3
1.3. 프로그램 구조	4
1.3.1. 시작 설정 부분	5
1.3.2. SQL 문장 실행 및 에러 처리 부분	6
1.3.3. 종료 설정 부분	7
제2장 데이터 타입	9
2.1. 개요	9
2.2. Tibero의 데이터 타입	9
2.3. tbCLI의 데이터 타입	11
제3장 tbCLI 함수	13
3.1. SQL-99 표준 API	13
3.1.1. SQLAllocConnect(Deprecated)	13
3.1.2. SQLAllocEnv(Deprecated)	13
3.1.3. SQLAllocHandle	13
3.1.4. SQLAllocStmt(Deprecated)	15
3.1.5. SQLBindCol	15
3.1.6. SQLBindParameter	17
3.1.7. SQLCancel	18
3.1.8. SQLCloseCursor	19
3.1.9. SQLColAttribute	20
3.1.10. SQLColAttributes(Deprecated)	22
3.1.11. SQLColumnPrivileges	22
3.1.12. SQLColumns	23
3.1.13. SQLConnect	26
3.1.14. SQLCopyDesc	28
3.1.15. SQLDescribeCol	29
3.1.16. SQLDescribeParam	31
3.1.17. SQLDisconnect	32
3.1.18. SQLDriverConnect	33
3.1.19. SQLEndTran	35
3.1.20. SQLError(Deprecated)	37
3.1.21. SQLExecDirect	37
3.1.22. SQLExecute	40

3.1.23.	SQLExtendedFetch(Deprecated)	42
3.1.24.	SQLExtProcRaiseError	42
3.1.25.	SQLExtProcRaiseErrorWithMsg	43
3.1.26.	SQLFetch	44
3.1.27.	SQLFetchScroll	45
3.1.28.	SQLForeignKeys	47
3.1.29.	SQLFreeConnect(Deprecated)	49
3.1.30.	SQLFreeEnv(Deprecated)	49
3.1.31.	SQLFreeHandle	50
3.1.32.	SQLFreeStmt	51
3.1.33.	SQLGetConnectAttr	52
3.1.34.	SQLGetConnectOption(Deprecated)	53
3.1.35.	SQLGetCursorName	53
3.1.36.	SQLGetData	54
3.1.37.	SQLGetDescField	56
3.1.38.	SQLGetDescRec	58
3.1.39.	SQLGetDiagField	59
3.1.40.	SQLGetDiagRec	60
3.1.41.	SQLGetEnvAttr	62
3.1.42.	SQLGetExtProcAllocMemory	63
3.1.43.	SQLGetExtProcConnect	64
3.1.44.	SQLGetFunctions	65
3.1.45.	SQLGetInfo	66
3.1.46.	SQLGetStmtAttr	67
3.1.47.	SQLGetStmtOption(Deprecated)	68
3.1.48.	SQLGetTypeInfo	68
3.1.49.	SQLMoreResults	70
3.1.50.	SQLNativeSql	71
3.1.51.	SQLNumParams	72
3.1.52.	SQLNumResultCols	73
3.1.53.	SQLParamData	74
3.1.54.	SQLParamOptions(Deprecated)	76
3.1.55.	SQLPrepare	76
3.1.56.	SQLPrimaryKeys	77
3.1.57.	SQLProcedureColumns	79
3.1.58.	SQLProcedures	80
3.1.59.	SQLPutData	82
3.1.60.	SQLRowCount	84
3.1.61.	SQLSetConnectAttr	85
3.1.62.	SQLSetConnectOption(Deprecated)	86
3.1.63.	SQLSetCursorName	86
3.1.64.	SQLSetDescField	87
3.1.65.	SQLSetDescRec	89

3.1.66.	SQLSetEnvAttr	91
3.1.67.	SQLSetParam(Deprecated)	92
3.1.68.	SQLSetPos	92
3.1.69.	SQLSetScrollOptions(Deprecated)	94
3.1.70.	SQLSetStmtAttr	94
3.1.71.	SQLSetStmtOption(Deprecated)	95
3.1.72.	SQLSpecialColumns	95
3.1.73.	SQLStatistics	97
3.1.74.	SQLTablePrivileges	99
3.1.75.	SQLTables	101
3.1.76.	SQLTransact(Deprecated)	102
3.2.	Tibero API	102
3.2.1.	SQLAllocEnv2(Deprecated)	102
3.2.2.	SQLAllocHandle2	103
3.2.3.	SQLGetWarningMsg	105
3.2.4.	SQLGetWarningMsgLength	105
3.2.5.	SQLLobClose	106
3.2.6.	SQLLobFreeLoc	107
3.2.7.	SQLLobGetData(Deprecated)	108
3.2.8.	SQLLobGetData2	110
3.2.9.	SQLLobGetLength	112
3.2.10.	SQLLobOpen	113
3.2.11.	SQLLobPutData	114
3.2.12.	SQLLobTruncate	116
제4장	tbCLI 에러 메시지	119
4.1.	반환 코드	119
4.2.	SQLSTATE	119
4.2.1.	01000 ~ 01007	120
4.2.2.	01S00 ~ 01S09	121
4.2.3.	07001 ~ 07009	122
4.2.4.	07S01	122
4.2.5.	08001 ~ 08007	123
4.2.6.	08S01	123
4.2.7.	21S01 ~ 21S02	124
4.2.8.	22001 ~ 22026	124
4.2.9.	23000, 24000, 25000	126
4.2.10.	25S01 ~ 25S03	126
4.2.11.	28000, 34000	127
4.2.12.	3C000 ~ 3F000	127
4.2.13.	40001 ~ 40003, 42000	128
4.2.14.	42S01 ~ 42S22, 44000	128
4.2.15.	HY000 ~ HY024	129

4.2.16.	HY090 ~ HY111	132
4.2.17.	HYC00	135
4.2.18.	HYT00 ~ HYT01	135
4.2.19.	IM001 ~ IM015	136
4.2.20.	00000, 02000	138
제5장	tbCLI와 ODBC	139
5.1.	ODBC	139
5.2.	tbCLI와 ODBC의 연동	139
5.2.1.	Windows 계열에서의 설치	139
5.2.2.	UNIX 계열에서의 설치	144
5.3.	tbdsn.tbr과 ODBC 데이터 원본 관리자	147
색인		149

그림 목차

[그림 1.1]	클라이언트/서버 환경	4
[그림 1.2]	tbCLI 프로그램의 구조	4
[그림 5.1]	Tibero 6 ODBC Driver 설치 확인	140
[그림 5.2]	사용자 DSN - 추가	141
[그림 5.3]	사용자 DSN - Tibero 6 ODBC Driver 선택	142
[그림 5.4]	사용자 DSN - 데이터베이스 접속 정보 입력(IP, PORT 방식)	142
[그림 5.5]	사용자 DSN - 데이터베이스 접속 정보 입력(SID 방식)	143
[그림 5.6]	사용자 DSN - 등록 완료	143

예 목차

[예 1.1]	tbCLI 프로그램의 시작 설정	5
[예 1.2]	tbCLI 프로그램의 SQL 문장 실행 - 직접 실행	6
[예 1.3]	tbCLI 프로그램의 SQL 문장 실행 - 준비된 실행	6
[예 1.4]	tbCLI 프로그램의 종료 설정	8
[예 2.1]	DATE_STRUCT, SQL_DATE_STRUCT	12
[예 2.2]	TIME_STRUCT, SQL_TIME_STRUCT	12
[예 2.3]	TIME_STAMP_STRUCT, SQL_TIMESTAMP_STRUCT	12

안내서에 대하여

안내서의 대상

본 안내서는 Tibero[®](이하 Tibero)에서 제공하는 tbCLI의 기본 개념과 프로그램 구조 등을 알고자 하는 데이터베이스 관리자(Database Administrator, 이하 DBA), 애플리케이션 프로그램 개발자를 대상으로 기술한다.

안내서의 전제 조건

본 안내서를 원활히 이해하기 위해서는 다음과 같은 사항을 미리 알고 있어야 한다.

- 데이터베이스의 이해
- RDBMS의 이해
- SQL의 이해

안내서의 제한 조건

본 안내서는 Tibero를 실무에 적용하거나 운용하는 데 필요한 모든 사항을 포함하고 있지 않다. 따라서 설치, 환경설정 등 운용 및 관리에 대해서는 각 제품 안내서를 참고하기 바란다.

참고

Tibero의 설치 및 환경설정에 관한 내용은 "Tibero 설치 안내서"를 참고한다.

안내서 구성

Tibero tbCLI 안내서는 총 5개의 장으로 이루어져 있다.

각 장의 주요 내용은 다음과 같다.

- 제1장: tbCLI 소개

tbCLI의 기본 개념과 구성요소, 프로그램의 구조를 소개한다.

- 제2장: 데이터 타입

tbCLI 프로그램에서 사용하는 데이터 타입을 기술한다.

- 제3장: tbCLI 함수

tbCLI가 지원하는 함수를 기술한다.

- 제4장: tbCLI 에러 메시지

tbCLI 프로그램을 실행하는 도중에 발생할 수 있는 에러를 기술한다.

- 제5장: tbCLI와 ODBC

tbCLI와 ODBC를 연동하는 방법을 기술한다.

안내서 규약

표기	의미
<<AaBbCc123>>	프로그램 소스 코드의 파일명
<Ctrl>+C	Ctrl과 C를 동시에 누름
[Button]	GUI의 버튼 또는 메뉴 이름
진하게	강조
" "(따옴표)	다른 관련 안내서 또는 안내서 내의 다른 장 및 절 언급
'입력항목'	화면 UI에서 입력 항목에 대한 설명
하이퍼링크	메일 계정, 웹 사이트
>	메뉴의 진행 순서
+----	하위 디렉터리 또는 파일 있음
----	하위 디렉터리 또는 파일 없음
<u>참고</u>	참고 또는 주의사항
<u>주의</u>	주의할 사항
[그림 1.1]	그림 이름
[예 1.1]	예제 이름
AaBbCc123	Java 코드, XML 문서
[command argument]	옵션 파라미터
< xyz >	'<'와 '>' 사이의 내용이 실제 값으로 변경됨
	선택 사항. 예) A B: A나 B 중 하나
...	파라미터 등이 반복되어서 나옴
\${ }	환경변수

안내서 변경이력

[illegible]

시스템 사용 환경

	요구 사항
Platform	HP-UX 11i (IA64)
	Solaris (Solaris 10)
	AIX (PPC 5L/AIX 5.3)
	GNU (X86, 64, IA64)
	Linux kernel 3.10 이상
	Windows(x86) 64bit
Hardware	최소 2.5GB 하드디스크 공간
	2GB 이상 메모리 공간
Compiler	PSM (C99 지원 필요)
	tbESQL/C (C99 지원 필요)

관련 안내서

안내서	설명
Tibero 설치 안내서	설치 시 필요한 시스템 요구사항과 설치 및 제거 방법을 기술한 안내서이다.
Tibero 애플리케이션 개발자 안내서	각종 애플리케이션 라이브러리를 이용하여 애플리케이션 프로그램을 개발하는 방법을 기술한 안내서이다.
Tibero External Procedure 안내서	External Procedure를 소개하고 이를 생성하고 사용하는 방법을 기술한 안내서이다.
Tibero JDBC 개발자 안내서	Tibero에서 제공하는 JDBC 기능을 이용하여 애플리케이션 프로그램을 개발하는 방법을 기술한 안내서이다.
Tibero tbESQL/C 안내서	C 프로그래밍 언어를 사용해 데이터베이스 작업을 수행하는 각종 애플리케이션 프로그램을 작성하는 방법을 기술한 안내서이다.
Tibero tbESQL/COBOL 안내서	COBOL 프로그래밍 언어를 사용해 데이터베이스 작업을 수행하는 각종 애플리케이션 프로그램을 작성하는 방법을 기술한 안내서이다.
Tibero tbPSM 안내서	저장 프러시저 모듈인 tbPSM의 개념과 문법, 구성요소를 소개하고, 프로그램을 작성하는 데 필요한 제어 구조, 복합 타입, 서브 프로그램, 패키지 와 SQL 문장을 실행하고 에러를 처리하는 방법을 기술한 안내서이다.
Tibero tbPSM 참조 안내서	저장 프러시저 모듈인 tbPSM의 패키지를 소개하고, 이러한 패키지에 포함된 각 프러시저와 함수의 프로토타입, 파라미터, 예제 등을 기술한 참조 안내서이다.
Tibero 관리자 안내서	Tibero의 동작과 주요 기능의 원활한 수행을 보장하기 위해 DBA가 알아야 할 관리 방법을 논리적 또는 물리적 측면에서 설명하고, 관리를 지원하는 각종 도구를 기술한 안내서이다.
Tibero tbAdmin 안내서	SQL/PSM 처리와 DBA를 위한 시스템 관리 기능을 제공하는 GUI 기반의 툴인 tbAdmin을 소개하고, 설치 및 사용 방법을 기술한 안내서이다.
Tibero 유틸리티 안내서	데이터베이스와 관련된 작업을 수행하기 위해 필요한 유틸리티의 설치 및 환경설정, 사용 방법을 기술한 안내서이다.
Tibero TAS 안내서	Tibero Active Cluster (TAS)를 사용해서 Tibero의 파일을 관리하고자 하는 관리자를 대상으로 기술한 안내서이다.

안내서	설명
Tibero 에러 참조 안내서	Tibero를 사용하는 도중에 발생할 수 있는 각종 에러의 원인과 해결 방법을 기술한 안내서이다.
Tibero 참조 안내서	Tibero의 동작과 사용에 필요한 초기화 파라미터와 데이터 사전, 정적 뷰, 동적 뷰를 기술한 참조 안내서이다.
Tibero SQL 참조 안내서	데이터베이스 작업을 수행하거나 애플리케이션 프로그램을 작성할 때 필요한 SQL 문장을 기술한 참조 안내서이다.
Tibero Spatial 참조 안내서	Tibero에서 Geometry 타입에 대한 설명과 Spatial 기능 관련 프러시저 함수 목록 및 사용 방법 등을 기술한 안내서이다.
Tibero TEXT 참조 안내서	Tibero의 제공하는 Text Index를 소개하고, Text Index를 생성하고 사용하는 방법을 기술하는 안내서이다.

참고 문헌

제품	안내서
ODBC	ODBC Programmer's Reference: http://msdn2.microsoft.com/en-us/library/ms714177.aspx
X/Open	X/Open CAE Specification Data Management: SQL Call Level Interface (CLI) ISBN: 1-85912-081-4 X/Open Document Number: C451

제1장 tbCLI 소개

본 장에서는 tbCLI의 기본 개념과 구성요소, 프로그램의 구조를 소개한다.

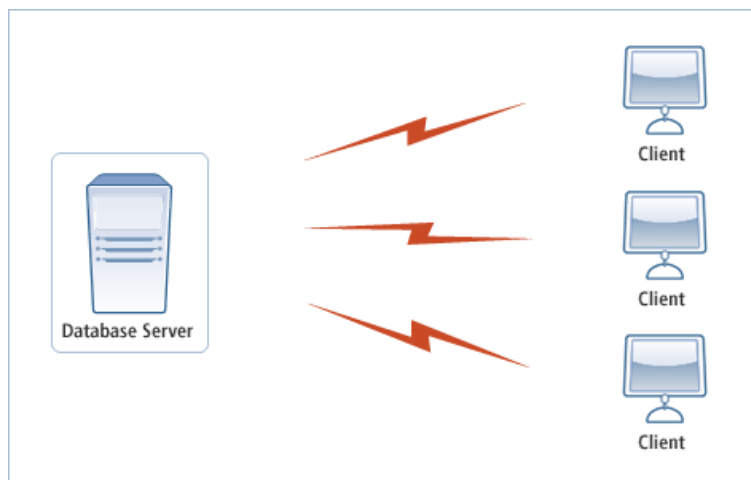
1.1. 개요

tbCLI는 Tibero가 제공하는 Call Level Interface(CLI)로 사용자의 애플리케이션 프로그램과 Tibero 간의 SQL 인터페이스 역할을 수행한다. 사용자는 tbCLI 라이브러리를 통해 Tibero에 접근하려는 C 또는 C++ 애플리케이션 프로그램을 쉽게 작성할 수 있다.

tbCLI는 ODBC(Open Database Connectivity) 및 X/Open Call Level Interface Standard를 기초로 개발되었다. tbCLI는 ODBC 2.0의 Level 2 및 ODBC 3.0의 Level 1의 모든 조건, 그리고 ODBC 3.0 Level 2의 대부분의 조건을 만족한다. 따라서 ODBC나 CLI를 이용해 작성된 기존의 애플리케이션 프로그램은 tbCLI 환경으로 쉽게 전환될 수 있다.

특히 tbCLI는 아래 그림과 같은 **클라이언트/서버** 환경에서 유용하다.

[그림 1.1] 클라이언트/서버 환경



클라이언트의 애플리케이션 프로그램이 tbCLI의 API를 호출하면, 데이터베이스 시스템이 이를 처리하고 그 결과를 클라이언트로 반환해 준다. 물론 사용자는 tbESQL 등의 인터페이스를 이용하여 데이터를 처리할 수 있지만, tbCLI를 사용함으로써 애플리케이션 프로그램과 데이터를 더 세밀하게 조작할 수 있다.

tbCLI는 다음과 같은 특징이 있다.

- 실행 파일을 생성할 때 프리컴파일러(Precompiler)가 필요 없다.

tbCLI는 tbESQL과 마찬가지로 일반적인 프로그램 언어와 SQL 문장의 장점을 융합한 인터페이스이다. 하지만 tbESQL과 다르게 일반적인 프로그램에 가깝다.

- 모듈을 효율적으로 관리할 수 있으며, 가독성이 뛰어나다.
- 애플리케이션 패키지를 바인딩할 필요가 없다.
- 접근할 데이터베이스의 통계(statistics)를 사용할 수 있다.
- 스레드의 안정성을 보장한다. 애플리케이션 프로그램을 멀티 스레드로 작성할 수 있도록 스레드의 안정성을 보장한다.

1.2. 구성요소

본 절에서는 tbCLI 프로그램을 작성하거나 실행하기 전에 알아야 할 구성요소에 대해 설명한다.

1.2.1. tbCLI 핸들

핸들은 tbCLI에서 관리하고 있는 몇 가지 주요 데이터 구조에 대한 포인터(pointer)이다. 핸들을 사용함으로써 tbCLI 프로그램은 복잡한 구조를 갖는 데이터를 보다 편리하게 관리할 수 있다.

핸들의 내부 데이터는 데이터 구조에 작업이 이루어지면 자동으로 갱신되기 때문에 애플리케이션 프로그램 개발자는 핸들이 가리키는 데이터 구조의 세부 사항에 대해 알 필요가 없으며, 내부 데이터를 직접 접근할 필요도 없다.

tbCLI 프로그램은 다음과 같이 네 가지 핸들을 사용한다.

- **환경 핸들(environment handle)**

tbCLI 프로그램의 환경에 대한 데이터를 포함한다. 포함되는 데이터로는 현재 환경의 상태, 환경에 할당되어 있는 연결 핸들의 리스트, 환경에 대한 에러 정보 등이다.

- **연결 핸들(connection handle)**

데이터소스(data source)와의 연결에 대한 데이터를 포함한다. 포함되는 데이터로는 현재 연결 상태, 연결에 할당되어 있는 문장 핸들의 리스트, 연결에 대한 에러 정보 등이다.

- **문장 핸들(statement handle)**

tbCLI 프로그램에서 실행할 하나의 SQL 문장에 대한 데이터를 포함한다. 포함되는 데이터로는 현재 문장 상태, 문장 내의 입력 파라미터 및 출력 컬럼의 리스트, 문장에 대한 에러 정보 등이다.

- **서술자 핸들(descriptor handle)**

문장 핸들과 연관된 결과 집합(ResultSet)의 각 컬럼이나 바인드된 파라미터의 데이터를 포함한다. 포함되는 데이터로는 SQL 문장의 입력 파라미터, 출력 컬럼에 대한 메타데이터(metadata) 등이다.

1.2.2. tbCLI 함수

tbCLI 프로그램에서 데이터베이스 작업을 수행하기 위해서는 tbCLI 함수를 사용해야 한다. 대부분의 tbCLI 함수는 대상이 되는 핸들을 입력 파라미터로 받으며, **SQLRETURN** 타입의 반환 코드를 갖는다.

다음은 tbCLI 함수 중에서 SQL 문장을 직접 실행할 수 있는 **SQLExecDirect** 함수의 프로토타입이다.

```
SQLRETURN SQLExecDirect(SQLHSTMT StatementHandle, SQLCHAR *SQLString,
                        SQLINTEGER SQLStringSize);
```

tbCLI 함수는 기능별로 몇 개의 그룹으로 나누어 제공된다. 예를 들면 핸들을 할당하거나 연결과 관련된 함수, SQL 문장 실행과 관련된 함수, SQL 질의 결과의 검색과 관련된 함수, 서술자 관련 함수, 에러 정보와 관련된 함수, 데이터소스 정보와 관련된 함수 등이 있다.

참고

tbCLI 함수의 사용에 대한 자세한 내용은 “제3장 tbCLI 함수”를 참고한다.

1.2.3. tbCLI 에러 메시지

tbCLI 프로그램은 tbCLI 함수를 실행한 후 반환되는 코드에 의해 실행 결과를 얻는다. 그러나 사용자는 보다 더 다양한 정보를 필요로 한다. 이러한 정보를 제공하기 위해서 tbCLI는 진단 레코드를 생성한다.

진단 레코드는 수행한 함수의 반환 코드뿐만 아니라 실행 결과에 대한 다양한 정보를 갖고 있다.

진단 레코드는 다음과 같이 두 가지 레코드로 구성된다.

- **헤더 레코드(header record)**

반환 코드, 로우 개수, 상태 레코드의 개수, 실행된 명령의 타입 등의 필드로 구성되어 있다.

반환 코드가 **SQL_INVALID_HANDLE**인 경우를 제외하고 tbCLI 함수가 실행되면 항상 헤더 레코드가 생성된다.

- **상태 레코드(status record)**

상태 레코드는 경고 및 에러에 대한 정보를 포함한다.

반환 코드가 **SQL_ERROR**, **SQL_SUCCESS_WITH_INFO**, **SQL_NO_DATA**, **SQL_NEED_DATA**, **SQL_STILL_EXECUTING**인 경우 생성된다.

상태 레코드의 필드 중에서 가장 중요한 필드 중의 하나는 **SQLSTATE** 필드이다. 에러 또는 경고 코드를 표준화한 것으로 값은 X/Open과 ISO/IEC 표준으로 정해져 있다. 형식은 **CCSSS** 다섯 자리의 문자 열이며, CC는 에러 클래스, SSS는 에러 서브 클래스를 의미한다.

진단 레코드는 하나의 헤더 레코드로 시작되고 경우에 따라 1개 이상의 상태 레코드가 추가된다. 진단 레코드의 값을 얻기 위해서는 **SQLGetDiagRec**와 **SQLGetDiagField** 함수를 사용해야 한다. 이 두 함수는 파라미터로 주어진 핸들에 포함된 진단 레코드의 정보를 반환하는 역할만을 수행한다.

함수	설명
SQLGetDiagField	진단 레코드 중 단일 필드의 정보를 얻기 위해 사용하는 함수이다. 헤더 레코드와 상태 레코드 모두에 사용할 수 있다.
SQLGetDiagRec	상태 레코드에 포함된 SQLSTATE 코드, 에러 코드, 진단 메시지 등의 여러 개의 필드 값을 동시에 얻기 위해 사용하는 함수이다. 상태 레코드에만 사용할 수 있다.

진단 레코드는 환경, 연결, 문장, 서술자 핸들에서 사용되고 관리된다.

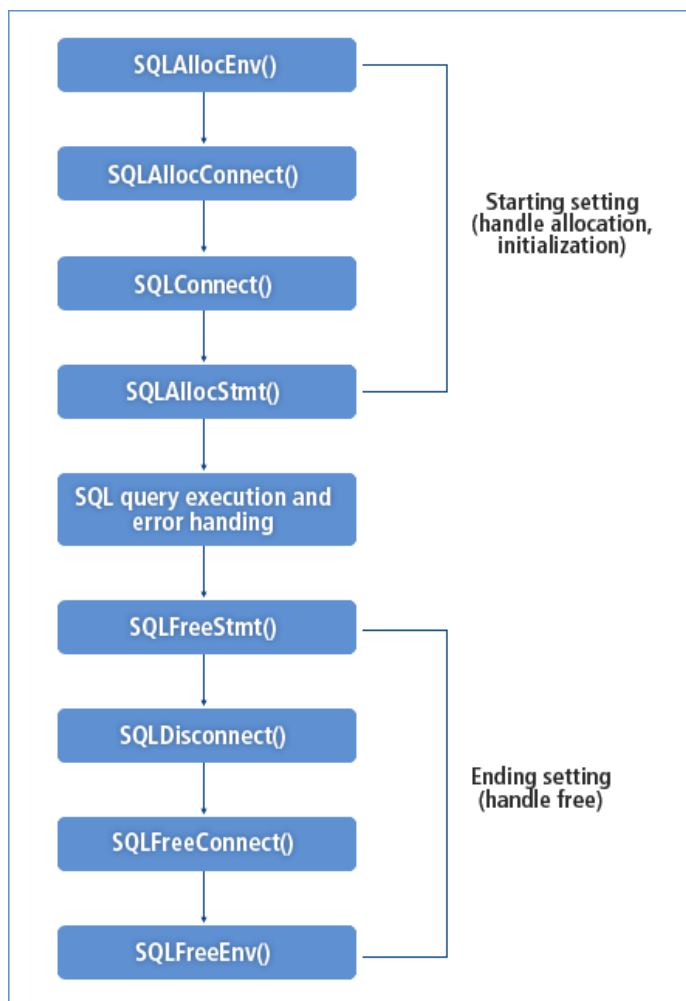
참고

에러 메시지에 대한 자세한 내용은 “제4장 **tbCLI 에러 메시지**”를 참고한다.

1.3. 프로그램 구조

tbCLI 프로그램은 아래 그림과 같이 크게 나눈다.

[그림 1.2] tbCLI 프로그램의 구조



- 시작 설정 부분(Starting setting)
- SQL 문장 실행 및 에러 처리 부분(SQL query execution and error handling)
- 종료 설정 부분(Ending setting)

1.3.1. 시작 설정 부분

tbCLI 프로그램을 시작하기 위해서는 우선 초기화 설정을 해야 한다. 초기화 설정을 하려면 환경 핸들과 연결 핸들을 할당하고 데이터소스와의 실제 연결을 수행해야 한다. 여기서 데이터소스란 Tibero의 소프트웨어 및 하드웨어의 전체 구성을 의미한다.

다음은 tbCLI 프로그램을 시작하는 예이다.

[예 1.1] tbCLI 프로그램의 시작 설정

```
SQLHENV h_env;

SQLHDBC h_dbc;

SQLRETURN rc = SQL_SUCCESS;

...

rc = SQLAllocHandle(SQL_HANDLE_ENV, NULL, &h_env);    ... ① ...

if (rc != SQL_SUCCESS)    ...

rc = SQLAllocHandle(SQL_HANDLE_DBC, h_env, &h_dbc);    ... ② ...

if (rc != SQL_SUCCESS)    ...

rc = SQLConnect(h_dbc, (SQLCHAR *)ds_name, SQL_NTS, (SQLCHAR *)user,
                SQL_NTS, (SQLCHAR *)passwd, SQL_NTS);    ... ③ ...

if (rc != SQL_SUCCESS && rc != SQL_SUCCESS_WITH_INFO)    ...
```

①, ② 초기화 설정을 위해 환경 핸들과 연결 핸들을 할당한다.

③ SQLConnect 함수를 통해 데이터소스에 연결한다.

이 함수를 호출할 때에는 파라미터로 데이터소스의 이름(ds_name)과 사용자의 이름(user), 비밀번호(passwd)를 함께 전달해야 한다. 뿐만 아니라 파라미터의 길이도 함께 설정해 주어야 하는데 위의 [예 1.1]에서는 길이 대신 NULL로 끝나는 문자열(null-terminating string) 즉 SQL_NTS를 설정한다.

④ 데이터소스에 연결하는 과정이 끝나면 tbCLI 프로그램은 SQL 문장을 실행하기 위해 다음과 같이 반드시 한 개 이상의 문장 핸들을 할당한다.

```
SQLHSTMT h_stmt;

...

rc = SQLAllocHandle(SQL_HANDLE_STMT, h_dbc, &h_stmt); ... ④ ...

if (rc != SQL_SUCCESS) ...
```

1.3.2. SQL 문장 실행 및 에러 처리 부분

SQL 문장을 실행하는 방법은 다음과 같이 두 가지가 있다.

직접 실행

SQL 문장을 `SQLExecDirect` 함수를 이용하여 한 번에 실행하는 방법이다.

다음은 직접 실행의 예이다.

[예 1.2] tbCLI 프로그램의 SQL 문장 실행 - 직접 실행

```
SQLCHAR *update = "UPDATE EMP SET SALARY = SALARY * 1.05 "
                  "WHERE DEPTNO = 5";

rc = SQLExecDirect(h_stmt, update, SQL_NTS);
if (rc != SQL_SUCCESS) ...
```

준비된 실행

`SQLPrepare`와 `SQLExecute` 함수를 이용하여 두 단계에 걸쳐 실행하는 방법이다.

대부분 SQL 문장 내에 파라미터가 포함된 경우에는 주로 준비된 실행 방법을 이용한다. `SQLPrepare`와 `SQLExecute` 함수 사이에 `SQLBindParameter` 함수를 호출하여 파라미터에 실제 값을 설정한다. 다음은 두 개의 입력 파라미터를 포함한 SQL 문장을 실행하는 예이다.

[예 1.3] tbCLI 프로그램의 SQL 문장 실행 - 준비된 실행

```
SQLCHAR *update = "UPDATE EMP SET SALARY = SALARY * ? "
                  "WHERE DEPTNO = ?";

double ratio = 0.0;
short deptno = 0;

...
```

```

rc = SQLPrepare(h_stmt, update, SQL_NTS);    ... ① ...

if (rc != SQL_SUCCESS) ...

rc = SQLBindParameter(h_stmt, 1, SQL_PARAM_INPUT, SQL_C_DOUBLE,
                      SQL_DOUBLE, 5, 2, &ratio, 0, NULL);    ... ㉠ ...

if (rc != SQL_SUCCESS) ...

rc = SQLBindParameter(h_stmt, 2, SQL_PARAM_INPUT, SQL_C_SHORT,
                      SQL_SMALLINT, 0, 0, &deptno, 0, NULL);    ... ㉡ ...

if (rc != SQL_SUCCESS) ...

ratio = 1.05;
deptno = 5;

SQLExecute(h_stmt);    ... ② ...

if (rc != SQL_SUCCESS) ...

```

위의 예에서는 SQL 문장 내의 입력 파라미터를 물음표(?)로 표시한다. 입력 파라미터의 위치를 표시할 때에는 1 이상의 정수를 사용한다.

① 준비된(prepared) SQL 문장에 포함된 각각의 입력 파라미터의 값을 저장하고 있는 변수의 포인터를 설정한다. 각각의 입력 파라미터(㉠, ㉡)에 입출력 방향, C 또는 C++의 데이터 타입, SQL의 데이터 타입, 정밀도(precision), 범위(scale) 등을 함께 설정한다.

② SQL 문장을 실행한다. 이 문장을 실행하면 설정된 입력 파라미터에 의해 EMP 테이블이 갱신된다.

SQL 문장을 실행하고 나서 몇 개의 로우가 갱신되었는지 확인하려면 SQLRowCount 함수를 사용한다.

사용하는 방법은 다음과 같다.

```
rc = SQLRowCount(h_stmt, &count);
```

각 함수의 반환 코드는 수행 결과에 대한 정보를 가지고 있으므로, 함수를 호출한 후에는 항상 반환 코드를 확인해야 한다.

1.3.3. 종료 설정 부분

tbCLI 프로그램을 종료하기 위해서는 “1.3.1. 시작 설정 부분”에서 수행한 작업과 반대되는 작업을 수행해야 한다.

다음은 tbCLI 프로그램을 종료하는 예이다.

[예 1.4] tbCLI 프로그램의 종료 설정

```
rc = SQLDisconnect(h_dbc);          ... ① ...  
  
if (rc != SQL_SUCCESS) ...  
  
SQLFreeHandle(SQL_HANDLE_DBC, h_dbc); ... ② ...  
  
SQLFreeHandle(SQL_HANDLE_ENV, h_env); ... ③ ...
```

① 데이터소스의 연결을 해제한다.

②, ③ 할당한 연결 핸들과 환경 핸들을 시스템에 반환한다.

제2장 데이터 타입

본 장에서는 tbCLI 프로그램에서 사용하는 데이터 타입을 알아보고 데이터 타입 간의 변환을 설명한다.

2.1. 개요

데이터 타입은 SQL 문장에 값을 입력하고, 질의 결과를 얻기 위해 사용한다.

tbCLI는 다음과 같이 두 가지 타입을 지원한다.

- **Tibero의 데이터 타입**

데이터베이스에 저장된 데이터에 접근할 때 사용한다.

- **tbCLI의 데이터 타입**

애플리케이션 프로그램에서 데이터를 조작할 때 사용한다.

2.2. Tibero의 데이터 타입

본 절에서는 Tibero에서 디폴트로 제공하는 데이터 타입을 간략히 설명한다. 이러한 데이터 타입은 데이터베이스의 스키마 객체를 생성하는 데 사용하는 것으로 tbESQL 프로그램 내에서도 모든 데이터 타입에 대응되는 변수를 사용할 수 있다.

다음은 Tibero의 데이터 타입이다.

구분	데이터 타입	설명
문자형	CHAR, VARCHAR, NCHAR, NVARCHAR, RAW, LONG, LONG RAW	문자열이나 바이너리 데이터를 저장하는 데이터 타입이다. LONG, LONG RAW의 경우 2GB까지 저장된다.
숫자형	NUMBER, INTEGER, FLOAT, BINARY_FLOAT, BINARY_DOUBLE	정수나 실수의 숫자를 저장하는 데이터 타입이다.
날짜형	DATE, TIME, TIMESTAMP, TIMESTAMP WITH TIME ZONE, TIMESTAMP WITH LOCAL TIMEZONE	시간이나 날짜, 시간대를 저장하는 데이터 타입이다.

구분	데이터 타입	설명
간격형	INTERVAL YEAR TO MONTH, INTERVAL DAY TO SECOND	시간사이의 간격을 저장하는 데이터 타입이다.
대용량 객체형	CLOB, BLOB	LOB 타입을 의미한다. 다른 데이터 타입이 지원하는 최대 길이(8KB 이하)보다 훨씬 큰 길이를 가질 수 있는 객체이다. 4GB까지 가능하다.
내재형	ROWID	사용자가 명시적으로 선언하지 않아도 Tibero가 자동으로 삽입되는 로우마다 포함하는 컬럼의 타입이다.

참고

자세한 내용은 "Tibero SQL 참조 안내서"를 참고한다.

다음은 각 데이터 타입에 대한 세부 설명이다.

데이터 타입	설명
CHAR VARCHAR	일반 문자열을 저장하는 데이터 타입이다.
NCHAR NVARCHAR	유니코드 문자열을 저장하는 데이터 타입이다.
RAW	바이너리 데이터를 저장하는 데이터 타입이다.
LONG LONG RAW	대용량의 문자열이나 바이너리 데이터를 저장하는 데이터 타입이다. 한 테이블 내에 하나의 컬럼만 선언할 수 있다.
NUMBER INTEGER FLOAT	정수 또는 실수를 저장하는 타입이다. NUMBER 타입을 선언할 때 정밀도와 스케일을 함께 선언할 수 있다. – 정밀도: 데이터 값의 전체 자릿수 – 스케일: 소수점 이하 자릿수
BINARY_FLOAT BINARY_DOUBLE	정수 또는 실수를 저장하는 타입이다. – BINARY_FLOAT: 32bit 부동 소수점 타입 – BINARY_DOUBLE: 64bit 부동 소수점 타입

데이터 타입	설명
DATE	특정 날짜와 시간, 시간대를 나타내는 데이터 타입이다.
TIME	– DATE: 특정 날짜
TIMESTAMP	– TIMESTAMP: 특정 날짜와 시간
TIMESTAMP WITH TIME ZONE	– TIMESTAMP WITH TIME ZONE: UTC(Coordinated Universal Time)으로 정규화된 특정 날짜와 시간, 해당 시간대
TIMESTAMP WITH LOCAL TIME ZONE	– TIMESTAMP WITH LOCAL TIME ZONE: UTC(Coordinated Universal Time)으로 정규화된 특정 날짜와 시간
INTERVAL YEAR TO MONTH	특정 날짜와 시간사이의 간격을 나타내는 데이터 타입이다.
INTERVAL DAY TO SECOND	– INTERVAL YEAR TO MONTH: 연도와 월로 계산한 시간 간격 – INTERVAL DAY TO SECOND: 일, 시, 분, 초, 소수점 초로 계산한 시간 간격
CLOB	대용량의 문자열이나 바이너리 데이터를 저장하는 데이터 타입이다.
BLOB	한 테이블 내에 여러 컬럼을 선언할 수 있다.
ROWID	데이터베이스 내의 각 로우를 식별하기 위해, Tiberio 시스템이 각 로우마다 자동으로 부여하는 데이터 타입이다. 각 로우가 저장되어 있는 물리적인 위치를 포함한다.

2.3. tbCLI의 데이터 타입

본 절에서는 애플리케이션 프로그램 개발자가 데이터베이스 프로그램을 작성할 때 사용하게 되는 tbCLI의 데이터 타입을 설명한다.

다음은 각 데이터 타입에 대한 C의 typedef 이름과 이에 대응하는 C의 데이터 타입을 보여주는 표이다.

C의 typedef 이름	C의 데이터 타입
SQLCHAR	unsigned char
SQLSCHAR	signed char
SQLSMALLINT	short int
SQLUSMALLINT	unsigned short int
SQLINTEGER	long int
SQLUINTEGER	unsigned long int
SQLREAL	float
SQLDOUBLE, SQLFLOAT	double

C의 typedef 이름	C의 데이터 타입
DATE_STRUCT, SQL_DATE_STRUCT	[예 2.1]
TIME_STRUCT, SQL_TIME_STRUCT	[예 2.2]
TIME_STAMP_STRUCT, SQL_TIMES TAMP_STRUCT	[예 2.3]

다음은 C의 typedef 이름에 대응하는 tbCLI의 데이터 타입을 사용한 예이다.

- DATE_STRUCT, SQL_DATE_STRUCT

[예 2.1] DATE_STRUCT, SQL_DATE_STRUCT

```
typedef struct tagDATE_STRUCT
{
    SQLSMALLINT    year;
    SQLUSMALLINT   month;
    SQLUSMALLINT   day;
};
```

- TIME_STRUCT, SQL_TIME_STRUCT

[예 2.2] TIME_STRUCT, SQL_TIME_STRUCT

```
struct tagTIME_STRUCT
{
    SQLUSMALLINT   hour;
    SQLUSMALLINT   minute;
    SQLUSMALLINT   second;
};
```

- TIME_STAMP_STRUCT, SQL_TIMESTAMP_STRUCT

[예 2.3] TIME_STAMP_STRUCT, SQL_TIMESTAMP_STRUCT

```
struct tagTIMESTAMP_STRUCT
{
    SQLSMALLINT    year;
    SQLUSMALLINT   month;
    SQLUSMALLINT   day;
    SQLUSMALLINT   hour;
    SQLUSMALLINT   minute;
    SQLUSMALLINT   second;
    SQLINTEGER      fraction;
};
```

제3장 tbCLI 함수

본 장에서는 tbCLI에서 지원하는 함수를 설명한다.

3.1. SQL-99 표준 API

3.1.1. SQLAllocConnect(Deprecated)

SQLAllocConnect 함수는 ODBC 3.x에서 [SQLAllocHandle](#) 함수로 대체된다.

3.1.2. SQLAllocEnv(Deprecated)

애플리케이션 프로그램은 한 번에 하나의 환경을 사용할 수 있다. 이 함수는 환경 핸들과 연관된 메모리를 할당하고 초기화한다.

환경 핸들을 할당받기 위해 SQLAllocConnect 등의 함수를 호출하기 전에 SQLAllocEnv 함수를 호출해야 한다. 그리고 호출한 이후에는 환경 핸들을 필요로 하는 모든 함수에 핸들이 전달된다.

참고로 SQLFreeEnv 함수를 호출하면 기존의 환경 핸들은 반환된다.

- 관련 함수

SQLAllocEnv 함수는 ODBC 3.x에서 [SQLAllocHandle](#) 함수로 대체된다.

3.1.3. SQLAllocHandle

환경, 연결, 문장 핸들을 할당할 때 사용하는 함수이다.

ISO 92 CLI/ODBC 3.0에서 지원하며, 자원, 환경, 명령문 핸들 초기화 및 메모리 할당을 수행하는 역할을 담당한다.

SQLAllocHandle 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLAllocHandle (SQLSMALLINT    HandleType,  
                           SQLHANDLE      InputHandle,  
                           SQLHANDLE      *OutputHandle);
```

- 파라미터

파라미터	용도	설명
HandleType	입력	할당 받을 핸들의 타입 – SQL_HANDLE_ENV – SQL_HANDLE_DBC – SQL_HANDLE_STMT
InputHandle	입력	입력 핸들(핸들의 타입에 따라 다른 값을 갖는다.) – SQL_HANDLE_ENV: SQL_NULL_HANDLE – SQL_HANDLE_DBC: 환경 핸들 – SQL_HANDLE_STMT: 연결 핸들
OutputHandle	출력	할당 받을 핸들의 포인터 (환경 핸들이 아닌 다른 핸들을 할당할 때 SQL_ERROR 를 반환하는 경우 OutputHandle 파라미터를 SQL_NULL_HDBC, SQL_NULL_HSTMT로 할당한다.)

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
08003	데이터베이스 연결이 존재하지 않는 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY009	NULL 포인터를 잘못 사용한 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY014	한정된 범위의 핸들 개수를 초과한 경우

SQLSTATE	설명
HY092	잘못된 속성 및 옵션 식별자
HYC00	선택적인 기능(Optional feature)이 구현되지 않은 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 <code>SQLAllocHandle</code> 함수를 지원하지 않는 경우

- 관련 함수

`SQLAllocHandle` 함수는 `SQLAllocConnect`, `SQLAllocEnv`, `SQLAllocstmt`를 대체하는 함수다.

ODBC 2.x 드라이버를 사용하는 애플리케이션 프로그램이 올바르게 동작하기 위해서는 `SQLAllocHandle` 함수를 `SQLAllocConnect`, `SQLAllocEnv`, `SQLAllocstmt` 함수로 적절하게 대체해야 한다.

3.1.4. SQLAllocStmt(Deprecated)

tbCLI에서 설명자, 결과 값, 커서 정보, 상태 정보를 처리하기 위해 사용하는 문장 핸들을 할당 받는 함수이다. 이 함수를 호출하면 문장을 위한 메모리를 할당하고 초기화한다. 하나의 데이터베이스 연결에 1,024개의 문장을 할당할 수 있다.

- 관련 함수

`SQLAllocStmt` 함수는 ODBC 3.x에서 `SQLAllocHandle` 함수로 대체된다.

3.1.5. SQLBindCol

ISO 92 CLI/ODBC 1.0에서 지원하며, 결과 집합의 컬럼을 받아올 버퍼와 데이터 타입을 설정하는 함수이다.

모든 데이터 타입에 대해 결과 집합의 컬럼을 애플리케이션 프로그램 변수에 연관시킨다. 이 함수에 의해 참조된 위치는 `SQLFetch` 함수를 호출하기 전까지 갱신되지 않으며 유효한 상태로 남아 있다.

`SQLFetch` 함수가 호출되면 데이터가 데이터베이스 서버에서 애플리케이션 프로그램으로 전송된다. 따라서 `SQLFetch` 함수를 사용하기 전에 이 함수를 호출해야 한다.

`SQLBindCol` 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLBindCol (SQLHSTMT      StatementHandle,
                      SQLUSMALLINT   ColumnNumber,
                      SQLSMALLINT     TargetType,
                      SQLPOINTER      TargetValue,
```

```
SQLINTEGER      BufferLength,  
SQLINTEGER      *StrLen_or_Ind);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
ColumnNumber	입력	컬럼 번호
TargetType	입력	C의 데이터 타입
TargetValue	입/출력	컬럼을 받아 올 버퍼의 포인터
BufferLength	입력	byte 단위의 버퍼 크기
StrLen_or_Ind	입/출력	버퍼에 저장된 데이터의 길이

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
07006	제한된 데이터 타입의 속성에 위배되는 경우
07009	잘못된 서술자 인덱스
HY000	일반적인 에러
HY001	메모리 할당 에러
HY003	잘못된 애플리케이션의 버퍼 타입
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY090	잘못된 문자열 또는 버퍼 길이
HYC00	선택적인 기능이 구현되지 않은 경우
HYT01	데이터베이스 연결 제한 시간이 만료된 경우
IM001	데이터베이스 연결 제한 시간을 초과한 경우

3.1.6. SQLBindParameter

ISO 92 CLI/ODBC 2.0에서 지원하며, SQL 문장에 파라미터를 바인드하는 함수이다. 즉, SQL 문장의 파라미터를 애플리케이션 프로그램의 변수에 바인드한다.

SQLBindParameter 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLBindParameter (SQLHSTMT      StatementHandle
                             SQLUSMALLINT  ParameterNumber,
                             SQLSMALLINT    InputOutputType,
                             SQLSMALLINT    ValueType,
                             SQLSMALLINT    ParameterType,
                             SQLULEN        ColumnSize,
                             SQLSMALLINT    DecimalDigits,
                             SQLPOINTER     ParameterValue,
                             SQLLEN         BufferLength,
                             SQLLEN         *StrLen_or_IndPtr);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
ParameterNumber	입력	파라미터의 위치
InputOutputType	입력	파라미터의 종류
ValueType	입력	C의 데이터 타입
ParameterType	입력	SQL의 데이터 타입
ColumnSize	입력	정밀도 - 데이터 타입이 다른 경우 • SQL_CHAR: 최대 길이 • SQL_NUMERIC: 최대 십진 정밀도
DecimalDigits	입력	SQL_NUMERIC인 경우 소수점 아래의 자릿수
ParameterValue	입/출력	파라미터의 포인터
BufferLength	입력	버퍼의 최대 크기
StrLen_or_IndPtr	입/출력	입/출력된 데이터의 크기

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
07006	제한된 데이터 타입의 속성에 위배되는 경우
07009	잘못된 서술자 인덱스
HY000	일반적인 에러
HY001	메모리 할당 에러
HY003	잘못된 애플리케이션의 버퍼 타입
HY004	잘못된 SQL의 데이터 타입
HY009	NULL 포인터를 잘못 사용한 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY021	모순이 있는(inconsistent) 서술자 정보
HY090	잘못된 문자열 또는 버퍼 길이
HY104	잘못된 정밀도 또는 스케일 값
HY105	잘못된 파라미터의 타입
HYC00	선택적인 기능이 구현되지 않은 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLBindParameter 함수를 지원하지 않는 경우

3.1.7. SQLCancel

ISO 92 CLI/ODBC 3.0에서 지원하며, 현재 실행 중인 SQL 문장의 수행을 취소시키는 함수이다.

SQLCancel 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLCancel (SQLHSTMT StatementHandle);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
HY000	일반적인 에러
HY001	메모리 할당 에러
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY018	서버에 취소 요청이 받아들여지지 않은 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLCancel 함수를 지원하지 않는 경우

3.1.8. SQLCloseCursor

ISO 92 CLI/ODBC 3.0에서 지원하며, 커서를 해제하는 함수이다. 즉, 현재 실행 중인 SQL 문장의 커서를 닫고 결과를 해제한다.

SQLCloseCursor 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLCloseCursor (SQLHSTMT StatementHandle);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
24000	잘못된 커서 상태인 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLCloseCursor 함수를 지원하지 않는 경우

3.1.9. SQLColAttribute

ISO 92 CLI/ODBC 3.0에서 지원하며, 결과 집합의 특정 컬럼에 속성을 설정하여 해당 정보를 확인하는 함수이다. 확인할 정보가 문자열이면 CharacterAttribute 파라미터에 반환되고, 숫자이면 NumericAttribute 파라미터에 반환된다.

이 함수를 호출하기 전에 SQLPrepare 또는 SQLExecDirect 함수를 호출해야 한다.

SQLColAttribute 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLColAttribute (SQLHSTMT      StatementHandle,
                           SQLUSMALLINT   ColumnNumber,
                           SQLUSMALLINT   FieldIdentifier,
                           SQLPOINTER      CharacterAttribute,
                           SQLSMALLINT     BufferLength,
                           SQLSMALLINT     *StringLength,
                           SQLPOINTER      NumericAttribute);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
ColumnNumber	입력	컬럼의 위치
FieldIdentifier	입력	확인할 정보
CharacterAttribute	출력	버퍼의 포인터
BufferLength	입력	버퍼의 최대 크기
StringLength	출력	반환되는 문자열의 길이
NumericAttribute	출력	반환되는 숫자

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_STILL_EXECUTING	이전에 실행한 SQL 문장이 완료되지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
01004	문자열 데이터가 오른쪽에서 잘린 경우
07005	커서의 범위가 아닌 준비된 문장인 경우
7009	잘못된 서술자 인덱스인 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY008	연산이 취소된 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY090	잘못된 문자열 또는 버퍼 길이
HY091	잘못된 서술자의 필드 식별자
HYC00	선택적인 기능이 구현되지 않은 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLColAttribute 함수를 지원하지 않는 경우

3.1.10. SQLColAttributes(Deprecated)

SQLColAttributes 함수는 ODBC 3.x에서 [SQLColAttribute](#) 함수로 대체된다.

3.1.11. SQLColumnPrivileges

ISO 92 CLI/ODBC 1.0에서 지원하며 특정 테이블의 컬럼 리스트와 관련된 권한을 검색하거나 반환하는 함수이다. 이때 반환된 정보는 결과 집합의 형태로 나타난다.

SQLColumnPrivileges 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLColumnPrivileges (SQLHSTMT          StatementHandle,
                                SQLCHAR            *CatalogName,
                                SQLSMALLINT         NameLength1,
                                SQLCHAR            *SchemaName,
                                SQLSMALLINT         NameLength2,
                                SQLCHAR            *TableName,
                                SQLSMALLINT         NameLength3,
                                SQLCHAR            *ColumnName,
                                SQLSMALLINT         NameLength4);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
CatalogName	입력	카탈로그 이름
NameLength1	입력	카탈로그 이름의 길이
SchemaName	입력	스키마 이름
NameLength2	입력	스키마 이름의 길이
TableName	입력	테이블 이름
NameLength3	입력	테이블 이름의 길이
ColumnName	입력	컬럼 이름
NameLength4	입력	컬럼 이름의 길이

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태

반환 코드	설명
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_STILL_EXECUTING	이전에 실행한 SQL 문장이 완료되지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
08S01	서버와의 통신 연결에 실패한 경우
24000	잘못된 커서 상태인 경우
40001	직렬화에 실패한 경우
40003	실행이 완료되지 않았고 트랜잭션의 상태를 알 수 없는 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY008	연산이 취소된 경우
HY009	NULL 포인터를 잘못 사용한 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY090	잘못된 문자열 또는 버퍼 길이
HYC00	선택적인 기능이 구현되지 않은 경우
HYT00	제한 시간을 초과한 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLColumnPrivileges 함수를 지원하지 않는 경우

3.1.12. SQLColumns

XopenCLI/ISO 92 CLI/ODBC 1.0에서 지원하며, 특정 테이블의 컬럼이나 리스트에 대한 정보를 검색하고 결과 집합을 반환하는 함수이다.

SQLColumns 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLColumns (SQLHSTMT      StatementHandle,
                      SQLCHAR        *CatalogName,
                      SQLSMALLINT    NameLength1,
```

```

SQLCHAR      *SchemaName,
SQLSMALLINT   NameLength2,
SQLCHAR      *TableName,
SQLSMALLINT   NameLength3,
SQLCHAR      *ColumnName,
SQLSMALLINT   NameLength4);

```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
CatalogName	입력	카탈로그 이름
NameLength1	입력	카탈로그 이름의 길이
SchemaName	입력	검색할 스키마의 이름
NameLength2	입력	스키마 이름의 길이
TableName	입력	검색할 테이블의 이름
NameLength3	입력	테이블 이름의 길이
ColumnName	입력	검색할 컬럼의 ID
NameLength4	입력	검색할 컬럼 ID의 길이

- 반환 컬럼

다음은 SQLColumns 함수에 의해 반환되는 컬럼이다.

컬럼 번호	컬럼 이름	데이터 타입	설명
1	TABLE_CAT	VARCHAR	카탈로그의 이름
2	TABLE_SCHEM	VARCHAR	스키마의 이름
3	TABLE_NAME	VARCHAR(NOT NULL)	테이블이나 뷰의 이름
4	COLUMN_NAME	VARCHAR(NOT NULL)	컬럼의 ID
5	DATA_TYPE	NUMERIC(NOT NULL)	컬럼에 대한 SQL의 데이터 타입
6	TYPE_NAME	VARCHAR(NOT NULL)	DATA_TYPE 컬럼을 나타내는 문자열
7	COLUMN_SIZE	NUMERIC	컬럼의 최대 크기
8	BUFFER_LENGTH	NUMERIC	데이터를 저장할 수 있는 최대 크기
9	DECIMAL_DIGITS	NUMERIC	컬럼의 소수 자릿수
10	NUM_PREC_RADIX	NUMERIC	10, 2, NULL 중의 하나의 값 – DATA_TYPE 컬럼이 숫자형인 경우: 2 – COLUMN_SIZE 컬럼: 십이제곱이 허용되는 비트의 수

컬럼 번호	컬럼 이름	데이터 타입	설명
11	NULLABLE	NUMERIC(NOT NULL)	NULL 값이 가능한지 여부
12	REMARKS	VARCHAR	컬럼의 설명
13	COLUMN_DEF	VARCHAR	컬럼의 디폴트 값
14	SQL_DATA_TYPE	NUMERIC(NOT NULL)	SQL의 데이터 타입
15	SQL_DATETIME_SUB	NUMERIC	내부 데이터 타입
16	CHAR_OCTET_LENGTH	NUMERIC	컬럼의 최대 자릿수
17	ORDINAL_POSITION	NUMERIC(NOT NULL)	테이블 컬럼의 원래 위치
18	IS_NULLABLE	VARCHAR	로우가 NULL를 포함하는지 여부

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_STILL_EXECUTING	이전에 실행한 SQL 문장이 완료되지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
08S01	서버와의 통신 연결에 실패한 경우
24000	잘못된 커서 상태인 경우
40001	직렬화에 실패한 경우
40003	실행이 완료되지 않았고 트랜잭션의 상태를 알 수 없는 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY008	연산이 취소된 경우
HY009	NULL 포인터를 잘못 사용한 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY090	잘못된 문자열 또는 버퍼 길이
HYC00	선택적인 기능이 구현되지 않은 경우
HYT00	제한 시간을 초과한 경우

SQLSTATE	설명
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLColumns 함수를 지원하지 않는 경우

3.1.13. SQLConnect

ISO 92 CLI/ODBC 1.0에서 지원하며, 연결할 대상이 되는 데이터베이스를 설정하는 함수이다. 애플리케이션 프로그램은 연결 대상인 데이터베이스에 SQL을 제공하고, 선택적으로 권한 및 인증과 관련된 정보를 제공할 수 있다.

이 함수를 호출하기 전에 SQLAllocConnect 함수를 호출해야 한다.

SQLConnect 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLConnect(SQLHDBC      ConnectionHandle,
                      SQLCHAR      *ServerName,
                      SQLSMALLINT   NameLength1,
                      SQLCHAR      *UserName,
                      SQLSMALLINT   NameLength2,
                      SQLCHAR      *Authentication,
                      SQLSMALLINT   NameLength3);
```

- 파라미터

파라미터	용도	설명
ConnectionHandle	입력	연결 핸들
ServerName	입력	데이터베이스 이름의 포인터
NameLength1	입력	데이터베이스 이름의 길이
UserName	입력	사용자 ID의 포인터
NameLength2	입력	사용자 ID의 길이
Authentication	입력	인증과 관련된 패스워드의 포인터
NameLength3	입력	인증과 관련된 패스워드의 길이

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태

반환 코드	설명
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
01S02	옵션의 값이 바뀐 경우
08001	클라이언트가 서버에 연결하는 것이 불가능한 경우
08002	이미 사용 중인 데이터베이스 연결의 이름인 경우
08004	서버가 연결을 거부한 경우
08S01	서버와의 통신 연결에 실패한 경우
28000	잘못된 권한을 설정한 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY013	메모리 관리 에러
HY090	잘못된 문자열 또는 버퍼 길이
HYT00	제한 시간을 초과한 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 <code>SQLConnect</code> 함수를 지원하지 않는 경우
IM002	데이터소스의 이름을 찾을 수 없거나, 디폴트 드라이버가 존재하지 않는 경우
IM003	설정된 드라이버를 로드할 수 없는 경우
IM004	해당 드라이버에 <code>SQLAllocHandle</code> 함수를 이용한 환경 핸들 할당에 실패한 경우
IM005	해당 드라이버에 <code>SQLAllocHandle</code> 함수를 이용한 연결 핸들 할당에 실패한 경우
IM006	드라이버에 <code>SQLSetConnectAttr</code> 함수를 이용한 데이터베이스 연결 속성 설정에 실패한 경우
IM009	Translation DLL에 연결하는 것이 불가능한 경우
IM010	데이터소스의 이름이 <code>SQL_MAX_DSN_LENGTH</code> 파라미터에 설정된 값 보다 긴 경우

3.1.14. SQLCopyDesc

ISO 92 CLI/ODBC 3.0에서 지원하며 다른 서술자 핸들에 서술자 정보를 복사하는 함수이다.

SQLCopyDesc 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLCopyDesc (SQLHDESC SourceDescHandle,  
                        SQLHDESC TargetDescHandle);
```

- 파라미터

파라미터	용도	설명
SourceDescHandle	입력	원본 서술자의 핸들
TargetDescHandle	입력	대상 서술자의 핸들

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
01S02	옵션의 값이 바뀐 경우
08001	클라이언트가 서버에 연결하는 것이 불가능한 경우
08002	이미 사용 중인 데이터베이스 연결의 이름인 경우
08004	서버가 연결을 거부한 경우
08S01	서버와의 통신 연결에 실패한 경우
28000	잘못된 권한을 설정한 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY013	메모리 관리 에러
HY090	잘못된 문자열 또는 버퍼 길이
HYT00	제한 시간을 초과한 경우

SQLSTATE	설명
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 <code>SQLCopyDesc</code> 함수를 지원하지 않는 경우
IM002	데이터소스의 이름을 찾을 수 없거나, 디폴트 드라이버가 존재하지 않는 경우
IM003	설정된 드라이버를 로드할 수 없는 경우
IM004	해당 드라이버에 <code>SQLAllocHandle</code> 함수를 이용한 환경 핸들의 할당에 실패한 경우
IM005	해당 드라이버에 <code>SQLAllocHandle</code> 함수를 이용한 연결 핸들의 할당에 실패한 경우
IM006	드라이버에 <code>SQLSetConnectAttr</code> 함수를 이용한 데이터베이스 연결 속성의 설정에 실패한 경우
IM009	Translation DLL에 연결하는 것이 불가능한 경우

3.1.15. SQLDescribeCol

ISO 92 CLI/ODBC 1.0에서 지원하며, 결과 집합에서 특정 컬럼의 이름, 데이터 타입, 정밀도, 스케일, NULL의 허용 여부 등의 정보를 확인하는 함수이다.

이 함수를 호출하기 전에 `SQLPrepare` 함수 또는 `SQLExecDirect` 함수를 호출해야 한다.

`SQLDescribeCol` 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLDescribeCol (SQLHSTMT      StatementHandle,
                          SQLUSMALLINT   ColumnNumber,
                          SQLCHAR         *ColumnName,
                          SQLSMALLINT     BufferLength,
                          SQLSMALLINT     *NameLength,
                          SQLSMALLINT     *DataType,
                          SQLUINTEGER     *ColumnSize,
                          SQLSMALLINT     *DecimalDigits,
                          SQLSMALLINT     *Nullable);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
ColumnNumber	입력	컬럼의 위치

파라미터	용도	설명
ColumnName	출력	컬럼 이름의 포인터
BufferLength	입력	컬럼 이름을 저장하기 위한 버퍼의 크기
NameLength	출력	컬럼 이름의 길이
DataType	출력	컬럼의 데이터 타입
ColumnSize	출력	컬럼의 정밀도
DecimalDigits	출력	컬럼의 스케일
Nullable	출력	컬럼에 대한 NULL의 허용 여부

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_STILL_EXECUTING	이전에 실행한 SQL 문장이 완료되지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
01004	문자열 데이터가 오른쪽에서 잘린 경우
07005	커서의 범위가 아닌 준비된 문장인 경우
07009	잘못된 서술자 인덱스
08S01	서버와의 통신 연결에 실패한 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY008	연산이 취소된 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY090	잘못된 문자열 또는 버퍼 길이
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLDescribeCol 함수를 지원하지 않는 경우

3.1.16. SQLDescribeParam

ODBC 1.0에서 지원하며, 준비된 SQL 문장과 관련된 파라미터 기호의 서술자를 반환하는 함수이다.

SQLDescribeParam 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLDescribeParam (SQLHSTMT      StatementHandle,  
                             SQLUSMALLINT  ParameterNumber,  
                             SQLSMALLINT    *DataType,  
                             SQLUINTEGER    *ParameterSize,  
                             SQLSMALLINT    *DecimalDigits,  
                             SQLSMALLINT    *Nullable);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
ParameterNumber	입력	파라미터 기호의 번호
DataType	출력	파라미터에 대한 SQL의 데이터 타입
ParameterSize	출력	컬럼의 크기가 반환될 버퍼의 포인터
DecimalDigits	출력	컬럼의 정밀도가 반환될 버퍼의 포인터
Nullable	출력	해당 파라미터에 대한 NULL의 허용 여부가 반환될 버퍼의 포인터 – SQL_NO_NULLS: NULL을 허용하지 않는다. – SQL_NULLABLE: NULL을 허용한다. – SQL_NULLABLE_UNKNOWN: NULL을 허용할 것인지를 결정할 수 없다.

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_STILL_EXECUTING	이전에 실행한 SQL 문장이 완료되지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
07009	잘못된 서술자 인덱스
08S01	서버와의 통신 연결에 실패한 경우
21S01	삽입될 값의 리스트가 컬럼과 일치하지 않은 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY008	연산이 취소된 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLDescribeParam 함수를 지원하지 않는 경우

3.1.17. SQLDisconnect

ISO 92 CLI/ODBC 1.0에서 지원하며, 연결 핸들과 관련된 데이터베이스 연결을 종료하는 함수이다. 이 함수가 호출되면 데이터베이스의 연결을 종료한 후 핸들을 해제한다. 성공적으로 핸들이 해제되고 나서 SQLConnect 함수를 호출하는 경우 애플리케이션 프로그램은 연결 핸들을 다시 사용할 수 있다.

이 함수를 호출한 후 다른 데이터베이스에 연결하려면 SQLConnect나 SQLFreeConnect 함수를 사용해야 한다.

SQLDisconnect 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLDisconnect (SQLHDBC ConnectionHandle);
```

- 파라미터

파라미터	용도	설명
ConnectionHandle	입력	연결 핸들

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태

반환 코드	설명
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
01002	데이터베이스 연결을 해제하는 중에 에러가 발생한 경우
08003	데이터베이스 연결이 존재하지 않는 경우
25000	잘못된 트랜잭션 상태인 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLDisconnect 함수를 지원하지 않는 경우

3.1.18. SQLDriverConnect

CLI/ODBC 1.0에서 지원하며, 연결할 대상이 되는 데이터베이스를 설정하는 함수이다. 이 함수는 SQLConnect 함수에서 제공하는 3개의 파라미터보다 더 많은 연결 정보를 필요로 하는 데이터소스를 지원한다.

이 함수를 호출하기 전에 SQLAllocConnect 함수를 호출해야 한다.

SQLDriverConnect 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLDriverConnect (SQLHDBC          ConnectionHandle,
                             SQLHWND          WindowHandle,
                             SQLCHAR          *InConnectionString,
                             SQLSMALLINT      StringLength1,
                             SQLCHAR          *OutConnectionString,
                             SQLSMALLINT      BufferLength,
                             SQLSMALLINT      *StringLength2Ptr,
                             SQLUSMALLINT     DriverCompletion);
```

- 파라미터

파라미터	용도	설명
ConnectionHandle	입력	연결 핸들

파라미터	용도	설명
WindowHandle	입력	사용하지 않음 (NULL 입력)
InConnectionString	입력	연결 문자열 연결 문자열은 keyword=value들로 구성되며 세미콜론(;)으로 구분한다. – Windows는 Keyword로 DSN, DESC(=Description), UID(=User), PWD(=Password), Server, Port, DB(=Database), SID, Connection Method, Option를 제공한다. – Unix 계열은 Keyword로 DSN, DESC(=Description), UID(=User), PWD(=Password), SID, Option을 제공한다.
StringLength1	입력	연결 문자열 길이
OutConnection String	출력	완성된 연결 문자열을 저장할 포인터 (최소 1024byte 크기의 버퍼를 주어야 하며, 여기에 NULL을 입력하면 BufferLength, StringLength2Ptr도 무시된다.)
BufferLength	입력	버퍼의 byte 크기
StringLength2Ptr	출력	버퍼에 저장된 데이터의 길이
DriverCompletion	입력	사용하지 않음 (SQL_DRIVER_NOPROMPT 입력)

● 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

● SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
01004	문자열 데이터가 오른쪽에서 잘린 경우
01S02	옵션의 값이 바뀐 경우
08001	클라이언트가 서버에 연결하는 것이 불가능한 경우
08002	이미 사용 중인 데이터베이스 연결의 이름인 경우
08004	서버가 연결을 거부한 경우
08S01	서버와의 통신 연결에 실패한 경우

SQLSTATE	설명
28000	잘못된 권한을 설정한 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY013	메모리 관리 에러
HY090	잘못된 문자열 또는 버퍼 길이
HYT00	제한 시간을 초과한 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 <code>SQLDriverConnect</code> 함수를 지원하지 않는 경우
IM002	데이터소스의 이름을 찾을 수 없거나, 디폴트 드라이버가 존재하지 않는 경우
IM003	설정된 드라이버를 로드할 수 없는 경우
IM004	해당 드라이버에 <code>SQLAllocHandle</code> 함수를 이용한 환경 핸들 할당에 실패한 경우
IM005	해당 드라이버에 <code>SQLAllocHandle</code> 함수를 이용한 연결 핸들 할당에 실패한 경우
IM006	드라이버에 <code>SQLSetConnectAttr</code> 함수를 이용한 데이터베이스 연결 속성 설정에 실패한 경우
IM009	Translation DLL에 연결하는 것이 불가능한 경우
IM010	데이터소스의 이름이 <code>SQL_MAX_DSN_LENGTH</code> 파라미터에 설정된 값 보다 긴 경우

3.1.19. SQLEndTran

ISO 92 CLI/ODBC 3.0에서 지원하며, 현재 트랜잭션을 커밋하거나 롤백하는 함수이다. 커밋과 롤백 중 어떤 동작을 실행할 것인지는 `SQLEndTran` 함수의 `CompletionType` 파라미터로 결정된다.

만약 현재 동작 중인 트랜잭션이 없는 경우에는 데이터베이스 서버에 영향을 주지 않고, `SQL_SUCCESS`를 반환한다.

`SQLEndTran` 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLEndTran (SQLSMALLINT   HandleType,
                      SQLHANDLE      Handle,
                      SQLSMALLINT   CompletionType);
```

- 파라미터

파라미터	용도	설명
HandleType	입력	할당 받을 핸들의 타입 – SQL_HANDLE_ENV – SQL_HANDLE_DBC
Handle	입력	핸들의 포인터
CompletionType	입력	커밋 또는 롤백 여부를 나타내는 타입 – SQL_COMMIT – SQL_ROLLBACK

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
08003	데이터베이스 연결이 존재하지 않는 경우
08007	트랜잭션을 수행하는 중에 데이터베이스 연결에 실패한 경우
25S01	트랜잭션의 상태
25S02	트랜잭션이 아직 활성화되지 않은 경우
25S03	트랜잭션이 롤백된 경우
40001	직렬화에 실패한 경우
40002	무결성 제약조건에 위배되는 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY010	함수 시퀀스 에러
HY012	잘못된 트랜잭션의 연산 코드인 경우
HY013	메모리 관리 에러
HY092	잘못된 속성 및 옵션 식별자

SQLSTATE	설명
HYC00	선택적인 기능이 구현되지 않은 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 <code>SQLEndTran</code> 함수를 지원하지 않는 경우

3.1.20. `SQLError`(Deprecated)

특정 명령문 및 데이터베이스 연결 또는 환경 핸들에 대해 가장 최근에 호출된 함수의 에러와 상태 정보를 반환하는 함수이다.

- 관련 함수

`SQLError` 함수는 ODBC 3.x에서 `SQLGetDiagRec` 함수로 대체된다.

3.1.21. `SQLExecDirect`

ISO 92 CLI/ODBC 1.0에서 지원하며, 커밋이나 롤백 등의 트랜잭션과 관련된 문장을 제외한 나머지 SQL 문장을 직접 실행하는 함수이다.

커밋이나 롤백을 실행하기 위해서는 `SQLTransact` 함수를 호출해야 한다. `SQLExecDirect` 함수를 호출하기 전에 모든 파라미터는 바인드되어야 하며, 연결할 데이터베이스 서버는 SQL 문장을 미리 준비해야 한다. 이때 SQL 문장은 한 번만 실행된다.

`SQLExecDirect` 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLExecDirect (SQLHSTMT StatementHandle,
                          SQLCHAR *StatementText,
                          SQLINTEGER TextLength);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
StatementText	입력	SQL 문장
TextLength	입력	SQL 문장의 텍스트 길이

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_NO_DATA	함수가 성공적으로 완료되었으나, 관련된 데이터를 찾을 수 없는 상태 (예: 주어진 SQL 문장에 의해 갱신된 로우가 없는 경우)
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_NEED_DATA	SQL 문장을 실행하기 위해 데이터가 더 필요한 상태 (예: 파라미터)
SQL_STILL_EXECUTING	이전에 실행한 SQL 문장이 완료되지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
01001	커서 연산이 충돌된 경우
01003	설정된 함수에서 NULL 값을 제거한 경우
01004	문자열 데이터가 오른쪽에서 잘린 경우
01006	회수되지 않은 특권
01007	부여되지 않은 특권
01S02	옵션의 값이 바뀐 경우
01S07	숫자 타입이나 날짜 타입의 일부 정보가 잘린 경우
07002	COUNT 필드에 오류가 있는 경우
07006	제한된 데이터 타입의 속성에 위배되는 경우
07S01	디폴트 파라미터를 잘못 사용한 경우
08S01	서버와의 통신 연결에 실패한 경우
21S01	삽입될 값의 리스트가 컬럼과 일치하지 않은 경우
21S02	테이블의 degree가 컬럼과 일치하지 않은 경우
22001	문자열 데이터가 오른쪽에서 잘린 경우
22002	지시자 변수가 요구되었으나 제공되지 않은 경우
22003	한정된 범위를 초과한 숫자 값인 경우
22007	잘못된 DATETIME 타입의 포맷인 경우
22008	DATETIME 필드에 오버플로우가 발생한 경우
22012	0으로 나눈 경우

SQLSTATE	설명
22015	INTERVAL 필드에 오버플로우가 발생한 경우
22018	캐스트 설정을 위한 문자 값이 잘못된 경우
22019	잘못된 이스케이프 문자
22025	잘못된 이스케이프 시퀀스
23000	무결성 제약조건에 위배되는 경우
24000	잘못된 커서 상태인 경우
34000	잘못된 커서 이름
3D000	잘못된 카탈로그 이름
3F000	잘못된 스키마 이름
40001	직렬화에 실패한 경우
40003	실행이 완료되지 않았고 트랜잭션의 상태를 알 수 없는 경우
42000	문법 에러 또는 접근 위배가 발생한 경우
42S01	기본 테이블 또는 뷰가 이미 존재하는 경우
42S02	기본 테이블 또는 뷰를 찾을 수 없는 경우
42S11	인덱스가 이미 존재하는 경우
42S12	인덱스를 찾을 수 없는 경우
42S21	컬럼이 이미 존재하는 경우
42S22	컬럼을 찾을 수 없는 경우
44000	WITH CHECK OPTION에 위배되는 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY008	연산이 취소된 경우
HY009	NULL 포인터를 잘못 사용한 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY090	잘못된 문자열 또는 버퍼 길이
HY105	잘못된 파라미터 타입
HY109	잘못된 커서 위치
HYC00	선택적인 기능이 구현되지 않은 경우
HYT00	제한 시간을 초과한 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLExecDirect 함수를 지원하지 않는 경우

3.1.22. SQLExecute

ISO 92 CLI/ODBC 1.0에서 지원하며, 준비된 SQL 문장을 한번 또는 여러 번 실행하는 함수이다. SQL 문장은 SQLBindParameter 함수를 사용하여 파라미터 마커(parameter maker)에 바인드된 애플리케이션 프로그램의 변수에 현재 값을 적용하여 실행된다.

애플리케이션 프로그램이 SQLExecute 함수를 사용하여 호출 결과를 처리하면, 애플리케이션 프로그램의 변수에 새로운 값을 대입하여 SQL 문장을 다시 실행할 수 있다.

SQLExecute 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLExecute (SQLHSTMT StatementHandle);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_NO_DATA	함수가 성공적으로 완료되었으나, 관련된 데이터를 찾을 수 없는 상태 (예: 주어진 SQL 문장에 의해 갱신된 로우가 없는 경우)
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_NEED_DATA	SQL 문장을 실행하기 위해 데이터가 더 필요한 상태 (예: 파라미터)
SQL_STILL_EXECUTING	이전에 실행한 SQL 문장이 완료되지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
01001	커서 연산이 충돌된 경우
01003	설정된 함수에서 NULL 값을 제거한 경우
01004	문자열 데이터가 오른쪽에서 잘린 경우

SQLSTATE	설명
01006	회수되지 않은 특권
01007	부여되지 않은 특권
01S02	옵션의 값이 바뀐 경우
01S07	숫자 타입이나 날짜 타입의 일부 정보가 잘린 경우
07002	COUNT 필드에 오류가 있는 경우
07006	제한된 데이터 타입의 속성에 위배되는 경우
07S01	디폴트 파라미터를 잘못 사용한 경우
08S01	서버와의 통신 연결에 실패한 경우
21S01	삽입될 값의 리스트가 컬럼과 일치하지 않은 경우
21S02	테이블의 degree가 컬럼과 일치하지 않은 경우
22001	문자열 데이터가 오른쪽에서 잘린 경우
22002	지시자 변수가 요구되었으나 제공되지 않은 경우
22003	한정된 범위를 초과한 숫자 값인 경우
22007	잘못된 DATETIME 타입의 포맷인 경우
22008	DATETIME 필드에 오버플로우가 발생한 경우
22012	0으로 나눈 경우
22015	INTERVAL 필드에 오버플로우가 발생한 경우
22018	캐스트 설정을 위한 문자 값이 잘못된 경우
22019	잘못된 이스케이프 문자
22025	잘못된 이스케이프 시퀀스
23000	무결성 제약조건에 위배되는 경우
24000	잘못된 커서 상태인 경우
40001	직렬화에 실패한 경우
40003	실행이 완료되지 않았고 트랜잭션의 상태를 알 수 없는 경우
42000	문법 에러 또는 접근 위배가 발생한 경우
44000	WITH CHECK OPTION에 위배되는 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY008	연산이 취소된 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY090	잘못된 문자열 또는 버퍼 길이

SQLSTATE	설명
HY105	잘못된 파라미터 타입
HY109	잘못된 커서 위치
HYC00	선택적인 기능이 구현되지 않은 경우
HYT00	제한 시간을 초과한 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 <code>SQLExecute</code> 함수를 지원하지 않는 경우

3.1.23. SQLExtendedFetch(Deprecated)

SQLExtendedFetch 함수는 ODBC 3.x에서 [SQLFetchScroll](#) 함수로 대체된다.

3.1.24. SQLExtProcRaiseError

External procedure 모듈을 위해서 지원하며, error를 발생시키는 함수이다.

SQLExtProcRaiseError 함수의 세부 내용은 다음과 같다.

- 문법

```
void SQL_API SQLExtProcRaiseError(ExtProcContext *Context,
    int ErrorCode);
```

- 파라미터

파라미터	용도	설명
Context	입력	컨텍스트
ErrorCode	입력	에러 코드

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
HY000	일반적인 에러
HY001	메모리 할당 에러

3.1.25. SQLExtProcRaiseErrorWithMsg

External procedure 모듈을 위해서 지원하며, 메시지와 함께 error를 발생시키는 함수이다.

SQLExtProcRaiseErrorWithMsg 함수의 세부 내용은 다음과 같다.

- 문법

```
void SQL_API SQLExtProcRaiseErrorWithMsg(ExtProcContext *Context,
                                         int      ErrorCode,
                                         char     *ErrorMessage);
```

- 파라미터

파라미터	용도	설명
Context	입력	컨텍스트
ErrorCode	입력	에러 코드
ErrorMessage	입력	에러 메시지

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
HY000	일반적인 에러
HY001	메모리 할당 에러

3.1.26. SQLFetch

ISO 92 CLI/ODBC 1.0에서 지원하며, SQL 문장의 실행 결과에 대해 페치(fetch)를 실행하고 선택된 모든 컬럼의 데이터를 반환하는 함수이다. 즉, 결과 집합 내에서 다음 행으로 커서를 이동하고 바인드된 컬럼을 가져온다. 바인드 되는 컬럼이 없는 경우 애플리케이션 프로그램으로 반환되는 데이터는 없으며 커서만 다음 행으로 이동된다. 그리고 커서가 이동한 후 바인드되지 않은 컬럼의 데이터는 삭제된다.

SQLFetch 함수는 가장 최근에 실행된 SQL 문장이 SELECT 문일 때만 호출할 수 있다.

SQLFetch 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLFetch (SQLHSTMT StatementHandle);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_NO_DATA	함수가 성공적으로 완료되었으나, 관련된 데이터를 찾을 수 없는 상태 (예: 주어진 SQL 문장에 의해 갱신된 로우가 없는 경우)
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_STILL_EXECUTING	이전에 실행한 SQL 문장이 완료되지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
01004	문자열 데이터가 오른쪽에서 잘린 경우
01S01	로우에 에러가 있는 경우
01S07	숫자 타입이나 날짜 타입의 일부 정보가 잘린 경우
07006	제한된 데이터 타입의 속성에 위배되는 경우
07009	잘못된 서술자 인덱스

SQLSTATE	설명
08S01	서버와의 통신 연결에 실패한 경우
22001	문자열 데이터가 오른쪽에서 잘린 경우
22002	지시자 변수가 요구되었으나 제공되지 않은 경우
22003	한정된 범위를 초과한 숫자 값인 경우
22007	잘못된 DATETIME 타입의 포맷인 경우
22012	0으로 나눈 경우
22015	INTERVAL 필드에 오버플로우가 발생한 경우
22018	캐스트 설정을 위한 문자 값이 잘못된 경우
24000	잘못된 커서 상태인 경우
40001	직렬화에 실패한 경우
40003	실행이 완료되지 않았고 트랜잭션의 상태를 알 수 없는 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY008	연산이 취소된 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY090	잘못된 문자열 또는 버퍼 길이
HY107	결과 집합 영역 내에서 존재하지 않는 로우 번호를 참조하려는 경우
HYC00	선택적인 기능이 구현되지 않은 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLFetch 함수를 지원하지 않는 경우

3.1.27. SQLFetchScroll

ISO 92 CLI/ODBC 3.0에서 지원하며, 결과 집합 내의 특정 로우의 집합을 페치하고 선택된 모든 컬럼을 반환하는 함수이다. 로우의 집합은 절대적 위치 또는 상대적 위치에 내용이 삽입된다.

SQLFetchScroll 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLFetchScroll (SQLHSTMT      StatementHandle,
                          SQLSMALLINT    FetchOrientation,
                          SQLINTEGER      FetchOffset);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
FetchOrientation	입력	스크롤 타입 - SQL_FETCH_NEXT: 다음 행 - SQL_FETCH_PRIOR: 이전 행 - SQL_FETCH_FIRST: 첫 번째 행 - SQL_FETCH_LAST: 마지막 행 - SQL_FETCH_ABSOLUTE: 시작점을 기준으로 FetchOffset 번째 행 - SQL_FETCH_RELATIVE: 현재 행을 기준으로 FetchOffset 번째 행
FetchOffset	입력	페치될 행의 번호

● 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_NO_DATA	함수가 성공적으로 완료되었으나, 관련된 데이터를 찾을 수 없는 상태 (예: 주어진 SQL 문장에 의해 갱신된 로우가 없는 경우)
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_STILL_EXECUTING	이전에 실행한 SQL 문장이 완료되지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

● SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
01004	문자열 데이터가 오른쪽에서 잘린 경우
01S01	로우에 에러가 있는 경우
01S06	결과 집합이 첫 번째 행의 집합을 반환하기 전에 페치를 시도한 경우
01S07	숫자 타입이나 날짜 타입의 일부 정보가 잘린 경우
07006	제한된 데이터 타입의 속성에 위배되는 경우
07009	잘못된 서술자 인덱스
08S01	서버와의 통신 연결에 실패한 경우
22001	문자열 데이터가 오른쪽에서 잘린 경우

SQLSTATE	설명
22002	지시자 변수가 요구되었으나 제공되지 않은 경우
22003	한정된 범위를 초과한 숫자 값인 경우
22007	잘못된 DATETIME 타입의 포맷인 경우
22012	0으로 나눈 경우
22015	INTERVAL 필드에 오버플로우가 발생한 경우
22018	캐스트 설정을 위한 문자 값이 잘못된 경우
24000	잘못된 커서 상태인 경우
40001	직렬화에 실패한 경우
40003	실행이 완료되지 않았고 트랜잭션의 상태를 알 수 없는 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY008	연산이 취소된 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY090	잘못된 문자열 또는 버퍼 길이
HY106	영역 밖의 페치 타입인 경우
HY107	결과 집합 영역 내에서 존재하지 않는 로우 번호를 참조하려는 경우
HY111	잘못된 북마크 값인 경우
HYC00	선택적인 기능이 구현되지 않은 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLFetchScroll 함수를 지원하지 않는 경우

3.1.28. SQLForeignKeys

ISO 92 CLI/ODBC 1.0에서 지원하며, 테이블의 외래 키에 대한 정보를 검색하는 함수이다. 즉, 다른 테이블 내의 기본 키를 참조하는 해당 테이블의 컬럼과 해당 테이블의 기본 키를 참조하는 다른 테이블의 외래 키의 목록을 검색한다.

SQLForeignKeys 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLForeignKeys (SQLHSTMT      StatementHandle,
                          SQLCHAR        *PKCatalogName,
                          SQLSMALLINT    NameLength1,
```

```

SQLCHAR      *PKSchemaName,
SQLSMALLINT   NameLength2,
SQLCHAR      *PKTableName,
SQLSMALLINT   NameLength3,
SQLCHAR      *FKCatalogName,
SQLSMALLINT   NameLength4,
SQLCHAR      *FKSchemaName,
SQLSMALLINT   NameLength5,
SQLCHAR      *FKTableName,
SQLSMALLINT   NameLength6);

```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
PKCatalogName	입력	기본 키의 카탈로그 이름
NameLength1	입력	기본 키의 카탈로그 이름의 길이
PKSchemaName	입력	기본 키의 스키마 이름
NameLength2	입력	기본 키의 스키마 이름의 길이
PKTableName	입력	기본 키의 테이블 이름
NameLength3	입력	기본 키의 테이블 이름의 길이
FKCatalogName	입력	외래 키의 카탈로그 이름
NameLength4	입력	외래 키의 카탈로그 이름의 길이
FKSchemaName	입력	외래 키의 스키마 이름
NameLength5	입력	외래 키의 스키마 이름의 길이
FKTableName	입력	외래 키의 테이블 이름
NameLength6	입력	외래 키의 테이블 이름의 길이

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_STILL_EXECUTING	이전에 실행한 SQL 문장이 완료되지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
08S01	서버와의 통신 연결에 실패한 경우
24000	잘못된 커서 상태인 경우
40001	직렬화에 실패한 경우
40003	실행이 완료되지 않았고 트랜잭션의 상태를 알 수 없는 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY008	연산이 취소된 경우
HY009	NULL 포인터를 잘못 사용한 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY090	잘못된 문자열 또는 버퍼 길이
HYC00	선택적인 기능이 구현되지 않은 경우
HYT00	제한 시간을 초과한 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLForeignKeys 함수를 지원하지 않는 경우

3.1.29. SQLFreeConnect(Deprecated)

연결 핸들을 무효화하고 이와 관련된 모든 자원을 해제하는 함수이다. 만약 데이터베이스 연결이 유지된 상태에서 SQLFreeConnect 함수를 호출하면, SQL_ERROR가 반환되고 연결 핸들은 유효해진다.

- 관련 함수

SQLFreeConnect 함수는 ODBC 3.x에서 [SQLDisconnect](#) 함수로 대체된다.

3.1.30. SQLFreeEnv(Deprecated)

환경 핸들을 무효화하고 이와 관련된 모든 자원을 해제하는 함수이다. 만약 연결 핸들이 유효한 상태에서 SQLFreeEnv 함수를 호출하면, SQL_ERROR가 반환되고 환경 핸들은 유효해진다.

이 함수를 호출하기 전에 SQLFreeConnect 함수를 호출해야 한다.

- 관련 함수

SQLFreeEnv 함수는 ODBC 3.x에서 [SQLFreeHandle](#) 함수로 대체된다.

3.1.31. SQLFreeHandle

ISO 92 CLI/ODBC 3.0에서 지원하며, 연결, 핸들, 명령문에 할당된 메모리를 해제하는 함수이다.

SQLFreeHandle 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLFreeHandle (SQLSMALLINT  HandleType,  
                          SQLHANDLE     Handle);
```

- 파라미터

파라미터	용도	설명
HandleType	입력	해제될 핸들의 타입 – SQL_HANDLE_ENV – SQL_HANDLE_DBC – SQL_HANDLE_STMT
Handle	입력	해제될 핸들이다.

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
HY000	일반적인 에러
HY001	메모리 할당 에러
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY017	자동으로 할당된 서술자 핸들을 잘못 사용한 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLFreeHandle 함수를 지원하지 않는 경우

3.1.32. SQLFreeStmt

ISO 92 CLI/ODBC 1.0에서 지원하며, 명령문에 할당된 메모리를 해제하는 함수이다.

SQL 문장을 실행한 후 결과를 처리하고 나서 `SQLFreeStmt` 함수를 호출한다. 이 함수는 열려있는 커서를 닫고, 파라미터를 재설정한다. 또한 컬럼의 바인드를 해제하고 문장 핸들을 무효화하며 핸들과 관련된 자원을 해제한다. 이때 `Option` 파라미터에 의해 핸들을 해제하는 방식을 설정할 수 있다.

`SQLFreeStmt` 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLFreeStmt (SQLHSTMT      StatementHandle,  
                        SQLUSMALLINT  Option);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
Option	입력	핸들을 해제하는 방식

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
HY000	일반적인 에러
HY001	메모리 할당 에러
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY092	잘못된 속성 및 옵션 식별자
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 <code>SQLFreeStmt</code> 함수를 지원하지 않는 경우

3.1.33. SQLGetConnectAttr

ISO 92 CLI/ODBC 3.0에서 지원하며, 현재 설정된 데이터베이스 연결의 속성을 반환하는 함수이다.

SQLGetConnectAttr 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLGetConnectAttr (SQLHDBC          ConnectionHandle,
                              SQLINTEGER       Attribute,
                              SQLPOINTER       Value,
                              SQLINTEGER       BufferLength,
                              SQLINTEGER       *StringLength);
```

- 파라미터

파라미터	용도	설명
ConnectionHandle	입력	연결 핸들
Attribute	입력	검색할 컬럼
Value	출력	컬럼의 현재 값을 저장한 메모리의 포인터
BufferLength	입력	실행 결과를 저장할 버퍼의 크기
StringLength	출력	Value 파라미터의 길이(NULL 포인터인 경우 길이를 반환하지 않는다.)

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_NO_DATA	함수가 성공적으로 완료되었으나, 관련된 데이터를 찾을 수 없는 상태 (예: 주어진 SQL 문장에 의해 갱신된 로우가 없는 경우)
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
01004	문자열 데이터가 오른쪽에서 잘린 경우
08003	데이터베이스 연결이 존재하지 않는 경우

SQLSTATE	설명
08S01	서버와의 통신 연결에 실패한 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY090	잘못된 문자열 또는 버퍼 길이
HY092	잘못된 속성 및 옵션 식별자
HYC00	선택적인 기능이 구현되지 않은 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLGetConnectAttr 함수를 지원하지 않는 경우

3.1.34. SQLGetConnectOption(Deprecated)

SQLGetConnectOption 함수는 ODBC 3.x에서 [SQLGetConnectAttr](#) 함수로 대체된다.

3.1.35. SQLGetCursorName

ISO 92 CLI/ODBC 1.0에서 지원하며, 명시된 문장과 관련된 커서의 이름을 반환하는 함수이다.

SQLGetCursorName 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLGetCursorName (SQLHSTMT      StatementHandle,
                             SQLCHAR        *CursorName,
                             SQLSMALLINT    BufferLength,
                             SQLSMALLINT    *NameLength);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
CursorName	출력	커서 이름을 저장한 버퍼의 포인터
BufferLength	입력	실행 결과를 저장할 버퍼의 크기
NameLength	출력	커서 이름의 길이

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
01004	문자열 데이터가 오른쪽에서 잘린 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY015	사용할 수 있는 커서의 이름이 없는 경우
HY090	잘못된 문자열 또는 버퍼 길이
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLGetCursorName 함수를 지원하지 않는 경우

3.1.36. SQLGetData

ISO 92 CLI/ODBC 1.0에서 지원하며, 결과 집합의 현재 로우에서 단일 컬럼에 대한 데이터를 검색하는 함수이다. SQLFetch 함수를 호출하여 애플리케이션 프로그램의 변수로 직접 데이터를 전달하는 SQLBindCol 함수 대신 이 함수를 사용할 수 있다.

SQLGetData 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLGetData (SQLHSTMT          StatementHandle,
                      SQLUSMALLINT      ColumnNumber,
                      SQLSMALLINT        TargetType,
                      SQLPOINTER         TargetValue,
                      SQLINTEGER          BufferLength,
                      SQLINTEGER          *StrLen_or_Ind);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
ColumnNumber	입력	검색할 컬럼의 위치
TargetType	입력	검색된 데이터가 삽입될 버퍼의 C의 타입
TargetValue	출력	검색된 데이터가 삽입될 버퍼의 포인터
BufferLength	입력	버퍼의 길이
StrLen_or_Ind	출력	검색된 데이터의 길이나 지시자

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_NO_DATA	함수가 성공적으로 완료되었으나, 관련된 데이터를 찾을 수 없는 상태 (예: 주어진 SQL 문장에 의해 갱신된 로우가 없는 경우)
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_STILL_EXECUTING	이전에 실행한 SQL 문장이 완료되지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
01004	문자열 데이터가 오른쪽에서 잘린 경우
01S07	숫자 타입이나 날짜 타입의 일부 정보가 잘린 경우
07006	제한된 데이터 타입의 속성에 위배되는 경우
07009	잘못된 서술자 인덱스
08S01	서버와의 통신 연결에 실패한 경우
22002	지시자 변수가 요구되었으나 제공되지 않은 경우
22003	한정된 범위를 초과한 숫자 값인 경우
22007	잘못된 DATETIME 타입의 포맷인 경우
22012	0으로 나눈 경우
22015	INTERVAL 필드에 오버플로우가 발생한 경우
22018	캐스트 설정을 위한 문자 값이 잘못된 경우
24000	잘못된 커서 상태인 경우

SQLSTATE	설명
HY000	일반적인 에러
HY001	메모리 할당 에러
HY003	잘못된 애플리케이션 버퍼의 타입
HY008	연산이 취소된 경우
HY009	NULL 포인터를 잘못 사용한 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY090	잘못된 문자열 또는 버퍼 길이
HY109	잘못된 커서 위치
HYC00	선택적인 기능이 구현되지 않은 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLGetData 함수를 지원하지 않는 경우

3.1.37. SQLGetDescField

ISO 92 CLI/ODBC 3.0에서 지원하며, 서술자 레코드의 컬럼 값 또는 현재 설정된 내용을 반환하는 함수이다.

SQLGetDescField 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLGetDescField (SQLHDESC      DescriptorHandle,
                           SQLSMALLINT    RecNumber,
                           SQLSMALLINT    FieldIdentifier,
                           SQLPOINTER     Value,
                           SQLINTEGER     BufferLength,
                           SQLINTEGER     *StringLength);
```

- 파라미터

파라미터	용도	설명
DescriptorHandle	입력	서술자 핸들
RecNumber	입력	서술자 레코드
FieldIdentifier	입력	서술자 값의 컬럼
Value	출력	서술자 정보를 저장한 버퍼의 포인터
BufferLength	입력	서술자 정보를 저장할 버퍼의 크기

파라미터	용도	설명
StringLength	출력	실행 결과 정보의 길이

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_NO_DATA	함수가 성공적으로 완료되었으나, 관련된 데이터를 찾을 수 없는 상태 (예: RecNumber 파라미터가 서술자 레코드의 현재 숫자보다 큰 경우, 서술자 핸들이 IRD인데 해당 커서가 열리지 않은 경우)
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
01004	문자열 데이터가 오른쪽에서 잘린 경우
07009	잘못된 서술자 인덱스
08S01	서버와의 통신 연결에 실패한 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY007	SQL 문장이 준비되지 않은 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY021	모순이 있는 서술자 정보
HY090	잘못된 문자열 또는 버퍼 길이
HY091	잘못된 서술자의 필드 식별자
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLGetDescField 함수를 지원하지 않는 경우

3.1.38. SQLGetDescRec

ISO 92 CLI/ODBC 3.0에서 지원하며, 서술자 레코드의 여러 개의 컬럼 값 또는 현재 설정된 내용을 반환하는 함수이다. 이때 컬럼은 명시된 이름, 데이터 타입 그리고 컬럼 및 파라미터의 데이터를 저장하는 공간을 반환한다.

SQLGetDescRec 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLGetDescRec (SQLHDESC      DescriptorHandle,
                          SQLSMALLINT   RecNumber,
                          SQLCHAR       *Name,
                          SQLSMALLINT   BufferLength,
                          SQLSMALLINT   *StringLength,
                          SQLSMALLINT   *Type,
                          SQLSMALLINT   *SubType,
                          SQLINTEGER    *Length,
                          SQLSMALLINT   *Precision,
                          SQLSMALLINT   *Scale,
                          SQLSMALLINT   *Nullable);
```

- 파라미터

파라미터	용도	설명
DescriptorHandle	입력	서술자 핸들
RecNumber	입력	서술자 레코드
Name	출력	SQL_DESC_NAME 컬럼을 저장한 버퍼의 포인터
BufferLength	입력	Name 파라미터의 버퍼 길이
StringLength	출력	반환된 SQL_DESC_NAME 컬럼의 길이
Type	출력	SQL_DESC_TYPE 컬럼을 저장한 버퍼의 포인터
SubType	출력	SQL_DATETIME 타입인 레코드에 대해 SQL_DESC_DATETIME_INTERVAL_CODE 필드의 값이 반환될 버퍼의 포인터
Length	출력	서술자 레코드의 길이 정보
Precision	출력	서술자 레코드의 정밀도 정보
Scale	출력	서술자 레코드의 스케일 정보
Nullable	출력	서술자 레코드의 NULL 허용 여부 정보

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태

반환 코드	설명
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_NO_DATA	함수가 성공적으로 완료되었으나, 관련된 데이터를 찾을 수 없는 상태 (예: RecNumber 파라미터가 서술자 레코드의 현재 숫자보다 큰 경우, 서술자 핸들이 IRD인데 해당 커서가 열리지 않은 경우)
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
01004	문자열 데이터가 오른쪽에서 잘린 경우
07009	잘못된 서술자 인덱스
08S01	서버와의 통신 연결에 실패한 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY007	SQL 문장이 준비되지 않은 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLGetDescRec 함수를 지원하지 않는 경우

3.1.39. SQLGetDiagField

ISO 92 CLI/ODBC 3.0에서 지원하며, 특정 핸들과 관련된 에러, 경고, 상태 정보를 포함하는 진단의 데이터 구조 내 임의의 필드의 현재 값을 반환하는 함수이다.

SQLGetDiagField 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLGetDiagField (SQLSMALLINT      HandleType,
                           SQLHANDLE          Handle,
                           SQLSMALLINT        RecNumber,
                           SQLSMALLINT        DiagIdentifier,
                           SQLPOINTER         DiagInfo,
```

SQLSMALLINT	BufferLength,
SQLSMALLINT	*StringLength);

- 파라미터

파라미터	용도	설명
HandleType	입력	핸들의 타입
Handle	입력	핸들
RecNumber	입력	상태 레코드
DiagIdentifier	입력	반환될 값의 진단 컬럼
DiagInfo	출력	진단 정보를 저장한 버퍼의 포인터
BufferLength	입력	버퍼의 길이
StringLength	출력	실행 결과 정보의 길이

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_NO_DATA	함수가 성공적으로 완료되었으나, 관련된 데이터를 찾을 수 없는 상태 (예: 주어진 SQL 문장에 의해 갱신된 로우가 없는 경우)
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
HY009	NULL 포인터를 잘못 사용한 경우
HY001	메모리 할당 에러

3.1.40. SQLGetDiagRec

ISO 92 CLI/ODBC 3.0에서 지원하며, 특정 핸들과 관련된 에러, 경고, 상태 정보를 포함하는 진단 레코드 내의 여러 필드의 현재 값을 반환하는 함수이다. 이 함수는 하나의 열을 반환하는 SQLGetDiagField 함수와는 다르게 SQLSTATE, 에러 코드, 진단 메시지의 텍스트 등을 반환한다.

SQLGetDiagRec 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLGetDiagRec (SQLSMALLINT      HandleType,
                        SQLHANDLE          Handle,
                        SQLSMALLINT      RecNumber,
                        SQLCHAR           *Sqlstate,
                        SQLINTEGER        *NativeError,
                        SQLCHAR           *MessageText,
                        SQLSMALLINT      BufferLength,
                        SQLSMALLINT      *TextLength);
```

- 파라미터

파라미터	용도	설명
HandleType	입력	핸들의 타입
Handle	입력	핸들
RecNumber	입력	상태 레코드
Sqlstate	출력	SQLSTATE 코드
NativeError	출력	디폴트 에러 코드
MessageText	출력	진단 메시지의 텍스트 문자열
BufferLength	입력	진단 메시지의 텍스트 문자열을 저장할 버퍼 길이
TextLength	출력	진단 메시지의 텍스트 문자열의 길이

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	함수가 성공적으로 완료되었으나, 관련된 데이터를 찾을 수 없는 상태 (예: RecNumber 파라미터가 0이거나 음수인 경우, 버퍼의 길이가 0보다 작은 경우)

- SQLSTATE

SQLSTATE	설명
HY009	NULL 포인터를 잘못 사용한 경우
HY001	메모리 할당 에러

3.1.41. SQLGetEnvAttr

ISO 92 CLI/ODBC 3.0에서 지원하며, 환경 핸들과 관련된 컬럼의 현재 값을 반환하는 함수이다.

SQLGetEnvAttr 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLGetEnvAttr (SQLHENV      EnvironmentHandle,
                          SQLINTEGER    Attribute,
                          SQLPOINTER    Value,
                          SQLINTEGER    BufferLength,
                          SQLINTEGER    *StringLength);
```

- 파라미터

파라미터	용도	설명
EnvironmentHandle	입력	환경 핸들
Attribute	입력	검색할 컬럼
Value	출력	컬럼의 현재 값
BufferLength	입력	Value 파라미터가 문자열인 경우 최대 길이
StringLength	출력	실행 결과 정보의 길이

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_NO_DATA	함수가 성공적으로 완료되었으나, 관련된 데이터를 찾을 수 없는 상태 (예: 주어진 SQL 문장에 의해 갱신된 로우가 없는 경우)
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
01004	문자열 데이터가 오른쪽에서 잘린 경우
HY000	일반적인 에러

SQLSTATE	설명
HY001	메모리 할당 에러
HY013	메모리 관리 에러
HY092	잘못된 속성 및 옵션 식별자
HYC00	선택적인 기능이 구현되지 않은 경우
IM001	드라이버가 SQLGetEnvAttr 함수를 지원하지 않은 경우

3.1.42. SQLGetExtProcAllocMemory

External procedure 모듈을 위해서 지원하며, statement의 handle을 extproc의 context로 부터 할당하는 함수이다.

SQLGetExtProcAllocMemory 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLPOINTER SQL_API SQLExtProcAllocMemory(ExtProcContext *Context,
                                           SQLINTEGER AllocSize);
```

- 파라미터

파라미터	용도	설명
Context	입력	컨텍스트
AllocSize	입력	할당 크기

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
HY000	일반적인 에러
HY001	메모리 할당 에러

3.1.43. SQLGetExtProcConnect

External procedure 모듈을 위해서 지원 하며, env, connection을 extproc의 context와 connect로 부터 할당 하는 함수이다.

SQLGetExtProcConnect 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQL_API SQLGetExtProcConnect(ExtProcContext *Context,
                                         SQLHENV      *EnvironmentHandle,
                                         SQLHDBC       *ConnectionHandle,
                                         SQLSMALLINT   *ErrorHandleType,
                                         SQLHANDLE     *ErrorHandle);
```

- 파라미터

파라미터	용도	설명
Context	입력	컨텍스트
EnvironmentHandle	출력	환경 핸들
ConnectionHandle	출력	연결 핸들
ErrorHandleType	출력	에러 핸들 타입
ErrorHandle	출력	에러 핸들

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
HY000	일반적인 에러
HY001	메모리 할당 에러

3.1.44. SQLGetFunctions

ISO 92 CLI/ODBC 1.0에서 지원하며, 드라이버가 특정 함수를 지원하는지의 여부를 알기 위해 사용하는 함수이다. 이 함수는 드라이버 관리자 내에서 사용할 수 있으며, 드라이버 내에도 사용할 수 있다. 만약 드라이버 내에서 이 함수를 사용하는 경우 드라이버 관리자는 드라이버 내의 함수를 호출하게 된다.

SQLGetFunctions 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLGetFunctions (SQLHDBC          ConnectionHandle,
                           SQLUSMALLINT      FunctionId,
                           SQLUSMALLINT      *Supported);
```

- 파라미터

파라미터	용도	설명
ConnectionHandle	입력	연결 핸들
FunctionId	입력	ODBC 함수를 호출할 미리 정의된 값
Supported	출력	질의한 함수가 지원되는지의 여부가 저장될 버퍼의 포인터 – SQL_TRUE – SQL_FALSE

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
08S01	서버와의 통신 연결에 실패한 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러

SQLSTATE	설명
HY095	영역 밖의 함수 타입인 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우

3.1.45. SQLGetInfo

ISO 92 CLI/ODBC 1.0에서 지원하며, 드라이버의 연결과 관련된 데이터소스의 일반적인 정보를 반환하는 함수이다.

SQLGetInfo 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLGetInfo (SQLHDBC          ConnectionHandle,
                      SQLUSMALLINT      InfoType,
                      SQLPOINTER         InfoValue,
                      SQLSMALLINT        BufferLength,
                      SQLSMALLINT        *StringLength);
```

- 파라미터

파라미터	용도	설명
ConnectionHandle	입력	연결 핸들
InfoType	입력	정보의 타입
InfoValue	출력	정보를 저장할 버퍼의 포인터
BufferLength	입력	버퍼의 길이
StringLength	출력	반환된 정보의 길이

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고

SQLSTATE	설명
01004	문자열 데이터가 오른쪽에서 잘린 경우
08003	데이터베이스 연결이 존재하지 않는 경우
08S01	서버와의 통신 연결에 실패한 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY013	메모리 관리 에러
HY024	잘못된 속성 값
HY090	잘못된 문자열 또는 버퍼 길이
HY096	영역 밖의 정보 타입
HYC00	선택적인 기능이 구현되지 않은 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLGetInfo 함수를 지원하지 않는 경우

3.1.46. SQLGetStmtAttr

ISO 92 CLI/ODBC 3.0에서 지원하며, SQL 문장의 속성에 대한 현재 값을 반환하는 함수이다.

SQLGetStmtAttr 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLGetStmtAttr (SQLHSTMT      StatementHandle,
                           SQLINTEGER     Attribute,
                           SQLPOINTER     Value,
                           SQLINTEGER     BufferLength,
                           SQLINTEGER     *StringLength);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
Attribute	입력	검색할 속성
Value	출력	명시된 속성의 현재 값이 반환될 버퍼의 포인터
BufferLength	입력	버퍼의 길이
StringLength	출력	반환된 결과의 길이

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
01004	문자열 데이터가 오른쪽에서 잘린 경우
24000	잘못된 커서 상태인 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY090	잘못된 문자열 또는 버퍼 길이
HY092	잘못된 속성 및 옵션 식별자
HY109	잘못된 커서 위치
HYC00	선택적인 기능이 구현되지 않은 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 <code>SQLGetStmtAttr</code> 함수를 지원하지 않는 경우

3.1.47. SQLGetStmtOption(Deprecated)

`SQLGetStmtOption` 함수는 ODBC 3.x에서 [SQLGetStmtAttr](#) 함수로 대체된다.

3.1.48. SQLGetTypeInfo

ISO 92 CLI/ODBC 1.0에서 지원하며, 데이터소스가 지원하는 데이터 타입의 정보를 반환하는 함수이다. 이때 정보는 SQL 문장을 실행한 결과 집합의 형식으로 반환된다.

애플리케이션 프로그램은 `ALTER TABLE`과 `CREATE TABLE` 문장에 이 함수를 호출하는 경우 결과 집합의 `TYPE_NAME` 컬럼으로 반환될 타입의 이름을 사용한다.

`SQLGetTypeInfo` 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLGetTypeInfo (SQLHSTMT      StatementHandle,
                          SQLSMALLINT    DataType);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
DataType	입력	SQL의 데이터 타입

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_STILL_EXECUTING	이전에 실행한 SQL 문장이 완료되지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
01S02	옵션의 값이 바뀐 경우
08S01	서버와의 통신 연결에 실패한 경우
24000	잘못된 커서 상태인 경우
40001	직렬화에 실패한 경우
40003	실행이 완료되지 않았고 트랜잭션의 상태를 알 수 없는 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY004	잘못된 SQL의 데이터 타입
HY008	연산이 취소된 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HYC00	선택적인 기능이 구현되지 않은 경우
HYT00	제한 시간을 초과한 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우

SQLSTATE	설명
IM001	드라이버가 SQLGetTypeInfo 함수를 지원하지 않는 경우

3.1.49. SQLMoreResults

ISO 92 CLI/ODBC 1.0에서 지원하며, SET, SELECT, UPDATE, INSERT, DELETE 문을 포함하는 SQL 문장이 추가적으로 실행 결과를 사용할 것인지를 결정하는 함수이다. 만약 추가적으로 실행 결과를 사용하는 경우에는 해당 결과를 초기화한다.

SQLMoreResults 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLMoreResults(SQLHSTMT StatementHandle);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_NO_DATA	함수가 성공적으로 완료되었으나, 관련된 데이터를 찾을 수 없는 상태 (예: 주어진 SQL 문장에 의해 갱신된 로우가 없는 경우)
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_STILL_EXECUTING	이전에 실행한 SQL 문장이 완료되지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
01S02	옵션의 값이 바뀐 경우
08S01	서버와의 통신 연결에 실패한 경우
40001	직렬화에 실패한 경우
40003	실행이 완료되지 않았고 트랜잭션의 상태를 알 수 없는 경우

SQLSTATE	설명
HY000	일반적인 에러
HY001	메모리 할당 에러
HY008	연산이 취소된 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLMoreResults 함수를 지원하지 않는 경우

3.1.50. SQLNativeSql

ODBC 1.0에서 지원하며, 드라이버에 의해 변경되는 SQL 문자열을 반환하는 함수이다. 이때 SQL 문장은 실행하지 않는다.

SQLNativeSql 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLNativeSql (SQLHDBC          ConnectionHandle,
                        SQLCHAR          *InStatementText,
                        SQLINTEGER       TextLength1,
                        SQLCHAR          *OutStatementText,
                        SQLINTEGER       BufferLength,
                        SQLINTEGER       *TextLength2Ptr);
```

- 파라미터

파라미터	용도	설명
ConnectionHandle	입력	연결 핸들
InStatementText	입력	변경될 SQL 문자열
TextLength1	입력	변경될 SQL 문자열의 길이
OutStatementText	출력	반환된 SQL 문자열
BufferLength	입력	반환된 SQL 문자열을 저장할 버퍼의 길이
TextLength2Ptr	출력	반환 결과의 길이

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태

반환 코드	설명
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
01004	문자열 데이터가 오른쪽에서 잘린 경우
08003	데이터베이스 연결이 존재하지 않는 경우
08S01	서버와의 통신 연결에 실패한 경우
22007	잘못된 DATETIME 타입의 포맷인 경우
24000	잘못된 커서 상태인 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY009	NULL 포인터를 잘못 사용한 경우
HY013	메모리 관리 에러
HY090	잘못된 문자열 또는 버퍼 길이
HY109	잘못된 커서 위치
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLNativeSql 함수를 지원하지 않는 경우

3.1.51. SQLNumParams

ODBC 1.0에서 지원하며, SQL 문장 내의 파라미터의 개수를 반환하는 함수이다.

SQLNumParams 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLNumParams (SQLHSTMT      StatementHandle,
                        SQLSMALLINT    *ParameterCountPtr);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들

파라미터	용도	설명
ParameterCountPtr	출력	반환 정보를 저장한 버퍼의 포인터

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_STILL_EXECUTING	이전에 실행한 SQL 문장이 완료되지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
08S01	서버와의 통신 연결에 실패한 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY008	연산이 취소된 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLNumParams 함수를 지원하지 않는 경우

3.1.52. SQLNumResultCols

ISO 92 CLI/ODBC 1.0에서 지원하며, 입력 문장 핸들과 관련된 결과 집합의 컬럼 수를 반환하는 함수이다. 가장 최근에 실행된 SQL 문장이 SELECT 문이 아닌 경우 0을 반환한다.

이 함수를 호출하기 전에 SQLPrepare 또는 SQLExecDirect 함수를 호출해야 한다.

SQLNumResultCols 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLNumResultCols (SQLHSTMT          StatementHandle,
                             SQLSMALLINT       *ColumnCount);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
ColumnCount	출력	결과를 저장할 포인터

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_STILL_EXECUTING	이전에 실행한 SQL 문장이 완료되지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
08S01	서버와의 통신 연결에 실패한 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY008	연산이 취소된 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLNumResultCols 함수를 지원하지 않는 경우

3.1.53. SQLParamData

ISO 92 CLI/ODBC 1.0에서 지원하며, SQL 문장을 실행하는 시간에 파라미터 데이터를 지원하는 함수이다. 이 함수는 SQLPutData 함수와 함께 사용한다.

SQLParamData 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLParamData (SQLHSTMT      StatementHandle,
                        SQLPOINTER      *Value);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
Value	출력	SQLBindParameter 함수의 ParameterValue 파라미터나 SQLBindCol 함수의 TargetValue 파라미터를 저장하기 위한 버퍼의 포인터

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_NO_DATA	함수가 성공적으로 완료되었으나, 관련된 데이터를 찾을 수 없는 상태 (예: 주어진 SQL 문장에 의해 갱신된 로우가 없는 경우)
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_NEED_DATA	SQL 문장을 실행하기 위해 데이터가 더 필요한 상태 (예: 파라미터)
SQL_STILL_EXECUTING	이전에 실행한 SQL 문장이 완료되지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
07006	제한된 데이터 타입의 속성에 위배되는 경우
08S01	서버와의 통신 연결에 실패한 경우
22026	잘못된 길이를 가진 문자열 데이터
40001	직렬화에 실패한 경우
40003	실행이 완료되지 않았고 트랜잭션의 상태를 알 수 없는 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY008	연산이 취소된 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLParamData 함수를 지원하지 않는 경우

3.1.54. SQLParamOptions(Deprecated)

SQLParamOptions 함수는 ODBC 3.x에서 [SQLSetStmtAttr](#) 함수로 대체된다.

3.1.55. SQLPrepare

ISO 92 CLI/ODBC 1.0에서 지원하며, 실행할 SQL 문장을 준비하는 함수이다. 즉, SQL 문장을 입력 명령 핸들과 연관시키고 명령문을 준비할 데이터베이스 서버로 전송한다.

SQLPrepare 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLPrepare (SQLHSTMT      StatementHandle,  
                      SQLCHAR        *StatementText,  
                      SQLINTEGER      TextLength);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
StatementText	입력	SQL 문장
TextLength	입력	SQL 문장의 길이

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_STILL_EXECUTING	이전에 실행한 SQL 문장이 완료되지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
01S02	옵션의 값이 바뀐 경우
08S01	서버와의 통신 연결에 실패한 경우
21S01	삽입될 값의 리스트가 컬럼과 일치하지 않은 경우
21S02	테이블의 degree가 컬럼과 일치하지 않은 경우

SQLSTATE	설명
22018	캐스트 설정을 위한 문자 값이 잘못된 경우
22019	잘못된 이스케이프 문자
22025	잘못된 이스케이프 시퀀스
24000	잘못된 커서 상태인 경우
34000	잘못된 커서 이름
3D000	잘못된 카탈로그 이름
3F000	잘못된 스키마 이름
42000	문법 에러 또는 접근 위배가 발생한 경우
42S01	기본 테이블 또는 뷰가 이미 존재하는 경우
42S02	기본 테이블 또는 뷰를 찾을 수 없는 경우
42S11	인덱스가 이미 존재하는 경우
42S12	인덱스를 찾을 수 없는 경우
42S21	컬럼이 이미 존재하는 경우
42S22	컬럼을 찾을 수 없는 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY008	연산이 취소된 경우
HY009	NULL 포인터를 잘못 사용한 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY090	잘못된 문자열 또는 버퍼 길이
HYC00	선택적인 기능이 구현되지 않은 경우
HYT00	제한 시간을 초과한 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLPrepare 함수를 지원하지 않는 경우

3.1.56. SQLPrimaryKeys

ISO 92 CLI/ODBC 1.0에서 지원하며, 특정 테이블에 대한 기본 키의 정보를 리스트 형태로 반환하는 함수이다.

SQLPrimaryKeys 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLPrimaryKeys (SQLHSTMT      StatementHandle,
                          SQLCHAR        *CatalogName,
                          SQLSMALLINT     NameLength1,
                          SQLCHAR        *SchemaName,
                          SQLSMALLINT     NameLength2,
                          SQLCHAR        *TableName,
                          SQLSMALLINT     NameLength3);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
CatalogName	입력	카탈로그 이름
NameLength1	입력	카탈로그 이름의 길이
SchemaName	입력	스키마 이름
NameLength2	입력	스키마 이름의 길이
TableName	입력	테이블 이름
NameLength3	입력	테이블 이름의 길이

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_STILL_EXECUTING	이전에 실행한 SQL 문장이 완료되지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
08S01	서버와의 통신 연결에 실패한 경우
24000	잘못된 커서 상태인 경우
40001	직렬화에 실패한 경우
40003	실행이 완료되지 않았고 트랜잭션의 상태를 알 수 없는 경우
HY000	일반적인 에러
HY001	메모리 할당 에러

SQLSTATE	설명
HY008	연산이 취소된 경우
HY009	NULL 포인터를 잘못 사용한 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY090	잘못된 문자열 또는 버퍼 길이
HYC00	선택적인 기능이 구현되지 않은 경우
HYT00	제한 시간을 초과한 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLPrimaryKeys 함수를 지원하지 않는 경우

3.1.57. SQLProcedureColumns

ODBC 1.0에서 지원하며, 명시된 프러시저의 결과 집합을 구성하는 컬럼뿐만 아니라, 입출력 파라미터의 목록을 반환하는 함수이다. 이때 정보는 명시된 문장에 대한 결과 집합의 형태로 반환한다.

SQLProcedureColumns 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLProcedureColumns (SQLHSTMT      StatementHandle,
                                SQLCHAR        *CatalogName,
                                SQLSMALLINT     NameLength1,
                                SQLCHAR        *SchemaName,
                                SQLSMALLINT     NameLength2,
                                SQLCHAR        *ProcName,
                                SQLSMALLINT     NameLength3,
                                SQLCHAR        *ColumnName,
                                SQLSMALLINT     NameLength4);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
CatalogName	입력	카탈로그 이름
NameLength1	입력	카탈로그 이름의 길이
SchemaName	입력	스키마 이름
NameLength2	입력	스키마 이름의 길이
ProcName	입력	프러시저 이름

파라미터	용도	설명
NameLength3	입력	프러시저 이름의 길이
ColumnName	입력	컬럼 이름
NameLength4	입력	컬럼 이름의 길이

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_STILL_EXECUTING	이전에 실행한 SQL 문장이 완료되지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
08S01	서버와의 통신 연결에 실패한 경우
24000	잘못된 커서 상태인 경우
40001	직렬화에 실패한 경우
40003	실행이 완료되지 않았고 트랜잭션의 상태를 알 수 없는 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY008	연산이 취소된 경우
HY009	NULL 포인터를 잘못 사용한 경우
HY010	함수 시퀀스 에러
HY090	잘못된 문자열 또는 버퍼 길이
HYC00	선택적인 기능이 구현되지 않은 경우
HYT00	제한 시간을 초과한 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLProcedureColumns 함수를 지원하지 않는 경우

3.1.58. SQLProcedures

ODBC 1.0에서 지원하며, 명시된 데이터소스 내에 저장된 프러시저 이름의 목록을 반환하는 함수이다.

SQLProcedures 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLProcedures (SQLHSTMT      StatementHandle,
                          SQLCHAR        *CatalogName,
                          SQLSMALLINT    NameLength1,
                          SQLCHAR        *SchemaName,
                          SQLSMALLINT    NameLength2,
                          SQLCHAR        *ProcName,
                          SQLSMALLINT    NameLength3);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
CatalogName	입력	카탈로그 이름
NameLength1	입력	카탈로그 이름의 길이
SchemaName	입력	스키마 이름
NameLength2	입력	스키마 이름의 길이
ProcName	입력	프러시저 이름
NameLength3	입력	프러시저 이름의 길이

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_STILL_EXECUTING	이전에 실행한 SQL 문장이 완료되지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
08S01	서버와의 통신 연결에 실패한 경우
24000	잘못된 커서 상태인 경우
40001	직렬화에 실패한 경우
40003	실행이 완료되지 않았고 트랜잭션의 상태를 알 수 없는 경우

SQLSTATE	설명
HY000	일반적인 에러
HY001	메모리 할당 에러
HY008	연산이 취소된 경우
HY009	NULL 포인터를 잘못 사용한 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY090	잘못된 문자열 또는 버퍼 길이
HYC00	선택적인 기능이 구현되지 않은 경우
HYT00	제한 시간을 초과한 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLProcedures 함수를 지원하지 않는 경우

3.1.59. SQLPutData

ISO 92 CLI/ODBC 1.0에서 지원하며, 애플리케이션 프로그램이 SQL 문장이 실행되는 시간에 파라미터 또는 컬럼의 데이터를 드라이버로 보내는 것을 허용하는 함수이다. 이 함수는 문자, 바이너리 또는 특정 데이터소스에서 사용하는 데이터 타입 등을 갖는 컬럼 내의 문자 또는 바이너리 데이터의 값을 보내는 데 사용한다.

현재 드라이버가 유니코드 데이터를 지원하지 않아도, 유니코드 C의 데이터 타입을 바인드할 수 있도록 지원한다.

SQLPutData 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLPutData (SQLHSTMT      StatementHandle,
                      SQLPOINTER      Data,
                      SQLINTEGER       StrLen_or_Ind);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
Data	입력	파라미터나 컬럼의 실제 데이터를 저장하는 버퍼의 포인터
StrLen_or_Ind	입력	데이터의 길이

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_STILL_EXECUTING	이전에 실행한 SQL 문장이 완료되지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
01004	문자열 데이터가 오른쪽에서 잘린 경우
07006	제한된 데이터 타입의 속성에 위배되는 경우
07S01	디폴트 파라미터가 잘못 사용된 경우
08S01	서버와의 통신 연결에 실패한 경우
22001	문자열 데이터가 오른쪽에서 잘린 경우
22003	한정된 범위를 초과한 숫자 값인 경우
22007	잘못된 DATETIME 타입의 포맷인 경우
22008	DATETIME 필드에 오버플로우가 발생한 경우
22012	0으로 나눈 경우
22015	INTERVAL 필드에 오버플로우가 발생한 경우
22018	캐스트 설정을 위한 문자 값이 잘못된 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY008	연산이 취소된 경우
HY009	NULL 포인터를 잘못 사용한 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY019	문자 및 바이너리가 아닌 데이터를 나누어 보낸 경우
HY020	NULL 값을 접합(union)하려고 시도하는 경우
HY090	잘못된 문자열 또는 버퍼 길이
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLPutData 함수를 지원하지 않는 경우

3.1.60. SQLRowCount

ISO 92 CLI/ODBC 1.0에서 지원하며, UPDATE, INSERT, DELETE 문에 의해 영향을 받는 테이블 또는 테이블과 관련된 뷰의 로우 수를 반환하는 함수이다. 이 함수는 가장 최근에 실행된 SQL 문장이 UPDATE, INSERT, DELETE 문이 아니거나 실행에 실패한 경우 0을 반환한다.

이 함수를 호출하기 전에 SQLExecute 함수 또는 SQLExecDirect 함수를 호출해야 한다.

SQLRowCount 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLRowCount (SQLHSTMT      StatementHandle,  
                        SQINTEGER      *RowCount);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
RowCount	출력	실행 결과의 로우를 저장할 포인터

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
HY000	일반적인 에러
HY001	메모리 할당 에러
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLRowCount 함수를 지원하지 않는 경우

3.1.61. SQLSetConnectAttr

ISO 92 CLI/ODBC 3.0에서 지원하며, 특정 데이터베이스 연결에 대한 연결 속성의 값을 설정하는 함수이다. 설정된 모든 데이터베이스 연결과 명령문의 옵션은 `SQLFreeConnect` 함수가 호출되거나 `SQLSetConnectAttr` 함수에 의해 재설정이 될 때까지 유지된다.

`SQLSetConnectAttr` 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLSetConnectAttr (SQLHDBC          ConnectionHandle,
                              SQLINTEGER       Attribute,
                              SQLPOINTER      Value,
                              SQLINTEGER      StringLength);
```

- 파라미터

파라미터	용도	설명
ConnectionHandle	입력	연결 핸들
Attribute	입력	설정할 속성
Value	입력	설정할 속성의 값이 들어 있는 버퍼의 포인터
StringLength	입력	버퍼의 길이

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
01S02	옵션의 값이 바뀐 경우
08002	이미 사용 중인 데이터베이스 연결의 이름인 경우
08003	데이터베이스 연결이 존재하지 않는 경우
08S01	서버와의 통신 연결에 실패한 경우
24000	잘못된 커서 상태인 경우
3D000	잘못된 카탈로그 이름

SQLSTATE	설명
HY000	일반적인 에러
HY001	메모리 할당 에러
HY009	NULL 포인터를 잘못 사용한 경우
HY010	함수 시퀀스 에러
HY011	현재 설정될 수 없는 속성인 경우
HY013	메모리 관리 에러
HY024	잘못된 속성 값
HY090	잘못된 문자열 또는 버퍼 길이
HY092	잘못된 속성 및 옵션 식별자
HYC00	선택적인 기능이 구현되지 않은 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLSetConnectAttr 함수를 지원하지 않는 경우
IM009	Translation DLL을 로드하는 것이 불가능한 경우

3.1.62. SQLSetConnectOption(Deprecated)

SQLSetConnectOption 함수는 ODBC 3.x에서 [SQLSetConnectAttr](#) 함수로 대체된다.

3.1.63. SQLSetCursorName

ISO 92 CLI/ODBC 1.0에서 지원하며, 현재 실행 중인 SQL 문장을 커서 이름으로 연관시키는 함수이다. 만약 애플리케이션 프로그램이 이 함수를 호출하지 않으면 드라이버가 SQL 문장을 실행하는 데 필요한 커서 이름을 생성한다.

SQLSetCursorName 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLSetCursorName (SQLHSTMT      StatementHandle,
                             SQLCHAR        *CursorName,
                             SQLSMALLINT    NameLength);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
CursorName	입력	커서 이름

파라미터	용도	설명
NameLength	입력	커서 이름의 길이

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
01004	문자열 데이터가 오른쪽에서 잘린 경우
24000	잘못된 커서 상태인 경우
34000	잘못된 커서 이름
3C000	중복된 커서 이름
HY000	일반적인 에러
HY001	메모리 할당 에러
HY009	NULL 포인터를 잘못 사용한 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY090	잘못된 문자열 또는 버퍼 길이
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLSetCursorName 함수를 지원하지 않는 경우

3.1.64. SQLSetDescField

ISO 92 CLI/ODBC 3.0에서 지원하며, 서술자 레코드의 한 필드의 값을 설정하는 함수이다.

SQLSetDescField 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLSetDescField (SQLHDESC      DescriptorHandle,
                           SQLSMALLINT    RecNumber,
```

SQLSMALLINT	FieldIdentifier,
SQLPOINTER	Value,
SQLINTEGER	BufferLength);

- 파라미터

파라미터	용도	설명
DescriptorHandle	입력	서술자 핸들
RecNumber	입력	서술자 내에서 해당 필드의 레코드 번호 (레코드 번호는 1부터 시작한다.)
FieldIdentifier	입력	설정될 필드의 식별자
Value	입력	설정할 값이 들어 있는 버퍼의 포인터
BufferLength	입력	버퍼의 길이

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
01S02	옵션의 값이 바뀐 경우
07009	잘못된 서술자 인덱스
08S01	서버와의 통신 연결에 실패한 경우
22001	문자열 데이터가 오른쪽에서 잘린 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY016	구현 로우 서술자(implementation row descriptor)를 수정할 수 없는 경우
HY021	모순이 있는 서술자 정보
HY090	잘못된 문자열 또는 버퍼 길이

SQLSTATE	설명
HY091	잘못된 서술자의 필드 식별자
HY092	잘못된 속성 및 옵션 식별자
HY105	잘못된 파라미터 타입
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLSetDescField 함수를 지원하지 않는 경우

3.1.65. SQLSetDescRec

ISO 92 CLI/ODBC 3.0에서 지원하며, 컬럼 또는 파라미터 데이터에 사용되는 버퍼와 데이터 타입에 대해 여러 개의 서술자 컬럼을 설정하는 함수이다.

SQLSetDescRec 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLSetDescRec (SQLHDESC      DescriptorHandle,
                          SQLSMALLINT   RecNumber,
                          SQLSMALLINT   Type,
                          SQLSMALLINT   SubType,
                          SQLINTEGER    Length,
                          SQLSMALLINT   Precision,
                          SQLSMALLINT   Scale,
                          SQLPOINTER    Data,
                          SQLINTEGER    *StringLength,
                          SQLINTEGER    *Indicator);
```

- 파라미터

파라미터	용도	설명
DescriptorHandle	입력	서술자 핸들
RecNumber	입력	설정할 컬럼을 포함하는 서술자 레코드
Type	입력	서술자 레코드의 종류
SubType	입력	서술자 레코드의 세부 종류
Length	입력	서술자 레코드의 길이 컬럼
Precision	입력	서술자 레코드의 정밀도 컬럼
Scale	입력	서술자 레코드의 스케일 컬럼
Data	입/출력	서술자 레코드의 데이터 포인터 컬럼
StringLength	입/출력	서술자 레코드의 문자열 길이 컬럼

파라미터	용도	설명
Indicator	입/출력	서술자 레코드의 지시자 포인터 컬럼

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
07009	잘못된 서술자 인덱스
08S01	서버와의 통신 연결에 실패한 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY016	구현 로우 서술자를 수정할 수 없는 경우
HY021	모순이 있는 서술자 정보
HY090	잘못된 문자열 또는 버퍼 길이
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLSetDescRec 함수를 지원하지 않는 경우

3.1.66. SQLSetEnvAttr

ISO 92 CLI/ODBC 3.0에서 지원하며, 현재 환경 핸들의 속성 값을 설정하는 함수이다.

SQLSetEnvAttr 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLSetEnvAttr (SQLHENV      EnvironmentHandle,
                          SQLINTEGER    Attribute,
                          SQLPOINTER    Value,
                          SQLINTEGER    StringLength);
```

- 파라미터

파라미터	용도	설명
EnvironmentHandle	입력	환경 핸들
Attribute	입력	설정할 환경 핸들의 속성
Value	입력	설정할 환경 핸들의 속성 값
StringLength	입력	설정할 환경 핸들의 속성 값이 문자형일 때 해당 길이

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
01S02	옵션의 값이 바뀐 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY009	NULL 포인터를 잘못 사용한 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY024	잘못된 속성 값

SQLSTATE	설명
HY090	잘못된 문자열 또는 버퍼 길이
HY092	잘못된 속성 및 옵션 식별자
HYC00	선택적인 기능이 구현되지 않은 경우

3.1.67. SQLSetParam(Deprecated)

SQLSetParam 함수는 ODBC 3.x에서 [SQLBindParameter](#) 함수로 대체된다.

3.1.68. SQLSetPos

ODBC 1.0에서 지원하며, 로우 집합 내에서 커서의 위치를 설정하거나 애플리케이션 프로그램이 로우 집합 내의 데이터를 리프레시(Refresh)하도록 허용하는 함수이다. 또한 이 함수는 결과 집합 내의 데이터를 갱신 또는 제거할 수 있다.

SQLSetPos 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLSetPos (SQLHSTMT      StatementHandle,
                     SQLUSMALLINT RowNumber,
                     SQLUSMALLINT Operation,
                     SQLUSMALLINT LockType);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
RowNumber	입력	로우 집합 내의 로우의 위치
Operation	입력	수행할 연산
LockType	입력	연산이 수행된 후 로우에 대한 잠금의 종류

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_NEED_DATA	SQL 문장을 실행하기 위해 데이터가 더 필요한 상태

반환 코드	설명
	(예: 파라미터)
SQL_STILL_EXECUTING	이전에 실행한 SQL 문장이 완료되지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

• SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
01001	커서 연산이 충돌한 경우
01004	문자열 데이터가 오른쪽에서 잘린 경우
01S01	로우에 에러가 있는 경우
01S07	숫자 타입이나 날짜 타입의 일부 정보가 잘린 경우
07006	제한된 데이터 타입의 속성에 위배되는 경우
07009	잘못된 서술자 인덱스
21S02	테이블의 degree가 컬럼과 일치하지 않은 경우
22001	문자열 데이터가 오른쪽에서 잘린 경우
22003	한정된 범위를 초과한 숫자 값인 경우
22007	잘못된 DATETIME 타입의 포맷인 경우
22008	DATETIME 필드에 오버플로우가 발생한 경우
22015	INTERVAL 필드에 오버플로우가 발생한 경우
22018	캐스트 설정을 위한 문자 값이 잘못된 경우
23000	무결성 제약조건에 위배되는 경우
24000	잘못된 커서 상태인 경우
40001	직렬화에 실패한 경우
40003	실행이 완료되지 않았고 트랜잭션의 상태를 알 수 없는 경우
42000	문법 에러 또는 접근 위배가 발생한 경우
44000	WITH CHECK OPTION에 위배되는 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY008	연산이 취소된 경우
HY010	함수 시퀀스 에러
HY011	현재 설정될 수 없는 속성인 경우
HY013	메모리 관리 에러

SQLSTATE	설명
HY090	잘못된 문자열 또는 버퍼 길이
HY092	잘못된 속성 식별자
HY107	결과 집합 영역 내에서 존재하지 않는 로우 번호를 참조하려는 경우
HY109	잘못된 커서 위치
HYC00	선택적인 기능이 구현되지 않은 경우
HYT00	제한 시간을 초과한 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLSetPos 함수를 지원하지 않는 경우

3.1.69. SQLSetScrollOptions(Deprecated)

SQLSetScrollOptions 함수는 ODBC 3.x에서 SQLGetInfo 함수와 SQLSetStmtAttr 함수로 대체된다.

3.1.70. SQLSetStmtAttr

ISO 92 CLI/ODBC 3.0에서 지원하며, 특정 문장 핸들의 속성을 설정하는 함수이다. 새로운 SQLSetStmtAttr 함수에 의해 속성이 재설정되거나 문장 핸들이 삭제되기 전에는 기존에 설정된 속성이 유지된다.

SQLSetStmtAttr 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLSetStmtAttr (SQLHSTMT      StatementHandle,
                           SQLINTEGER     Attribute,
                           SQLPOINTER     Value,
                           SQLINTEGER     StringLength);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
Attribute	입력	설정할 문장 핸들의 속성
Value	입력	설정할 문장 핸들의 속성 값
StringLength	입력	설정할 문장 핸들의 속성 값이 문자형일 때 해당 길이

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
01S02	옵션의 값이 바뀐 경우
08S01	서버와의 통신 연결에 실패한 경우
24000	잘못된 커서 상태인 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY009	NULL 포인터를 잘못 사용한 경우
HY010	함수 시퀀스 에러
HY011	현재 설정될 수 없는 속성인 경우
HY013	메모리 관리 에러
HY017	자동으로 할당된 서술자 핸들을 잘못 사용한 경우
HY024	잘못된 속성 값
HY090	잘못된 문자열 또는 버퍼 길이
HY092	잘못된 속성 및 옵션 식별자
HYC00	선택적인 기능이 구현되지 않은 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLSetStmtAttr 함수를 지원하지 않는 경우

3.1.71. SQLSetStmtOption(Deprecated)

SQLSetStmtOption 함수는 ODBC 3.x에서 [SQLSetStmtAttr](#) 함수로 대체된다.

3.1.72. SQLSpecialColumns

Xopen CLI/ISO 92 CLI/ODBC 1.0에서 지원하며, 테이블의 특정 컬럼의 정보를 검색하는 함수이다.

검색되는 컬럼의 정보로는 해당 테이블 내의 하나의 로우를 유일하게 설정하는 컬럼의 최적 집합(optimal set) 그리고 트랜잭션이 로우의 임의의 값을 변경할 때 자동으로 변경되는 컬럼이 있다.

SQLSpecialColumns 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLSpecialColumns (SQLHSTMT      StatementHandle,
                             SQLUSMALLINT   IdentifierType,
                             SQLCHAR         *CatalogName,
                             SQLSMALLINT     NameLength1,
                             SQLCHAR         *SchemaName,
                             SQLSMALLINT     NameLength2,
                             SQLCHAR         *TableName,
                             SQLSMALLINT     NameLength3,
                             SQLUSMALLINT    Scope,
                             SQLUSMALLINT    Nullable);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
IdentifierType	입력	반환되는 컬럼의 타입
CatalogName	입력	카탈로그 이름
NameLength1	입력	카탈로그 이름의 길이
SchemaName	입력	스키마 이름
NameLength2	입력	스키마 이름의 길이
TableName	입력	테이블 이름
NameLength3	입력	테이블 이름의 길이
Scope	입력	ROWID의 최소 범위
Nullable	입력	NULL 값의 허용 여부

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_STILL_EXECUTING	이전에 실행한 SQL 문장이 완료되지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
08S01	서버와의 통신 연결에 실패한 경우
24000	잘못된 커서 상태인 경우
40001	직렬화에 실패한 경우
40003	실행이 완료되지 않았고 트랜잭션의 상태를 알 수 없는 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY008	연산이 취소된 경우
HY009	NULL 포인터를 잘못 사용한 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY090	잘못된 문자열 또는 버퍼 길이
HY097	영역 밖의 컬럼인 경우
HY098	영역 밖의 범위(scope) 타입인 경우
HY099	영역 밖의 Nullable 타입인 경우
HYC00	선택적인 기능이 구현되지 않은 경우
HYT00	제한 시간을 초과한 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLSpecialColumns 함수를 지원하지 않는 경우

3.1.73. SQLStatistics

Xopen CLI/ODBC 1.0에서 지원하며, 특정 테이블의 인덱스 통계 정보를 검색하는 함수이다. 이때 정보는 결과 집합의 형태로 반환된다.

SQLStatistics 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLStatistics (SQLHSTMT      StatementHandle,
                        SQLCHAR          *CatalogName,
                        SQLSMALLINT      NameLength1,
                        SQLCHAR          *SchemaName,
                        SQLSMALLINT      NameLength2,
```

```
SQLCHAR      *TableName,
SQLSMALLINT  NameLength3,
SQLUSMALLINT Unique,
SQLUSMALLINT Reserved);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
CatalogName	입력	카탈로그 이름
NameLength1	입력	카탈로그 이름의 길이
SchemaName	입력	스키마 이름
NameLength2	입력	스키마 이름의 길이
TableName	입력	테이블 이름
NameLength3	입력	테이블 이름의 길이
Unique	입력	인덱스의 종류
Reserved	입력	SQL_ENSURE로 설정된 경우 데이터베이스에서 무조건 통계정보를 가져온다. 이 외에는 사용할 수 없다.

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_STILL_EXECUTING	이전에 실행한 SQL 문장이 완료되지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
1000	일반적인 경고
08S01	서버와의 통신 연결에 실패한 경우
24000	잘못된 커서 상태인 경우
40001	직렬화에 실패한 경우
40003	실행이 완료되지 않았고 트랜잭션의 상태를 알 수 없는 경우
HY000	일반적인 에러
HY001	메모리 할당 에러

SQLSTATE	설명
HY008	연산이 취소된 경우
HY009	NULL 포인터를 잘못 사용한 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY090	잘못된 문자열 또는 버퍼 길이
HY100	영역 밖의 고유 옵션 타입(uniqueness option type)
HY101	영역 밖의 정확도 옵션 타입(accuracy option type)
HYC00	선택적인 기능이 구현되지 않은 경우
HYT00	제한 시간을 초과한 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLStatistics 함수를 지원하지 않는 경우

3.1.74. SQLTablePrivileges

ISO 92 CLI/ODBC 1.0에서 지원하며, 각 테이블과 관련된 테이블이나 권한 목록의 정보를 반환하는 함수이다. 이때 정보는 결과 집합의 형태로 반환한다.

SQLTablePrivileges 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLTablePrivileges (SQLHSTMT      StatementHandle,
                               SQLCHAR        *CatalogName,
                               SQLSMALLINT     NameLength1,
                               SQLCHAR        *SchemaName,
                               SQLSMALLINT     NameLength2,
                               SQLCHAR        *TableName,
                               SQLSMALLINT     NameLength3);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
CatalogName	입력	카탈로그 이름
NameLength1	입력	카탈로그 이름의 길이
SchemaName	입력	스키마 이름
NameLength2	입력	스키마 이름의 길이

파라미터	용도	설명
TableName	입력	테이블 이름
NameLength3	입력	테이블 이름의 길이

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_STILL_EXECUTING	이전에 실행한 SQL 문장이 완료되지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
08S01	서버와의 통신 연결에 실패한 경우
24000	잘못된 커서 상태인 경우
40001	직렬화에 실패한 경우
40003	실행이 완료되지 않았고 트랜잭션의 상태를 알 수 없는 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY008	연산이 취소된 경우
HY009	NULL 포인터를 잘못 사용한 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY090	잘못된 문자열 또는 버퍼 길이
HYC00	선택적인 기능이 구현되지 않은 경우
HYT00	제한 시간을 초과한 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLTablePrivileges 함수를 지원하지 않는 경우

3.1.75. SQLTables

Xopen CLI/ISO 92 CLI/ODBC 1.0에서 지원하며, 연결된 특정 데이터소스의 데이터 사전에 저장된 테이블 이름의 리스트의 정보를 반환하는 함수이다. 이때 정보는 결과 집합의 형태로 반환되므로 **SELECT** 문에 의해 생성된 결과 집합과 같이 사용된다.

SQLTables 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLTables (SQLHSTMT      StatementHandle,
                      SQLCHAR        *CatalogName,
                      SQLSMALLINT     NameLength1,
                      SQLCHAR        *SchemaName,
                      SQLSMALLINT     NameLength2,
                      SQLCHAR        *TableName,
                      SQLSMALLINT     NameLength3,
                      SQLCHAR        *TableType,
                      SQLSMALLINT     NameLength4);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
CatalogName	입력	카탈로그 이름
NameLength1	입력	카탈로그 이름의 길이
SchemaName	입력	스키마 이름
NameLength2	입력	스키마 이름의 길이
TableName	입력	테이블 이름
NameLength3	입력	테이블 이름의 길이
TableType	입력	테이블의 데이터 타입
NameLength4	입력	테이블의 데이터 타입의 길이

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_STILL_EXECUTING	이전에 실행한 SQL 문장이 완료되지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
08S01	서버와의 통신 연결에 실패한 경우
24000	잘못된 커서 상태인 경우
40001	직렬화에 실패한 경우
40003	실행이 완료되지 않았고 트랜잭션의 상태를 알 수 없는 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY008	연산이 취소된 경우
HY009	NULL 포인터를 잘못 사용한 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY090	잘못된 문자열 또는 버퍼 길이
HYC00	선택적인 기능이 구현되지 않은 경우
HYT00	제한 시간을 초과한 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLTables 함수를 지원하지 않는 경우

3.1.76. SQLTransact(Deprecated)

수행된 데이터베이스의 모든 변경에 대해 커밋을 하거나 롤백을 하는 함수이다. 만약 트랜잭션이 사용 중이면 애플리케이션 프로그램은 데이터베이스의 연결이 끊기기 전에 이 함수를 호출해야 한다. 이 함수가 호출되면 열린 상태인 커서가 닫히고 처리되지 않은 결과 집합은 삭제된다.

- 관련 함수

SQLTransact 함수는 ODBC 3.x에서 [SQLEndTran](#) 함수로 대체된다.

3.2. Tiber API

3.2.1. SQLAllocEnv2(Deprecated)

[SQLAllocEnv](#) 함수와 같이 환경 핸들과 연관된 메모리를 할당하고 초기화하는 함수이다. 또한 [tbCLI](#)를 사용하는 애플리케이션 프로그램이 멀티 스레드를 사용하지는지의 여부를 설정할 수 있다.

멀티 스레드는 운영체제별로 설정하는 방법이 다음과 같이 다르다.

운영체제	설명
UNIX 계열	UNIX 계열은 디폴트로 멀티 스레드를 사용하지 않는 것으로 설정되어 있다. 멀티 스레드를 사용하려면 반드시 마지막 파라미터인 UseMultiThread의 값에 'TRUE'를 설정해줘야 한다.
Windows 계열	UseMultiThread의 값에 상관없이 멀티 스레드를 사용할 수 있으므로 <code>SQLAllocEnv2</code> 또는 <code>SQLAllocHandle2</code> 함수를 사용하지 않기를 권장한다.

이 함수는 환경 핸들을 처음 할당할 때 사용해야만 멀티스레드의 사용 여부를 적용할 수 있다.

- 관련 함수

`SQLAllocEnv2` 함수는 [SQLAllocHandle2](#) 함수로 대체된다.

3.2.2. SQLAllocHandle2

환경, 연결, 문장 핸들을 할당하고, `tbCLI`를 사용하는 프로그램이 멀티 스레드를 사용하는지의 여부를 설정할 수 있는 함수이다. 멀티 스레드를 운영체제 별로 설정하는 방법은 [SQLAllocHandle](#) 함수와 같다.

`SQLAllocHandle2` 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLAllocHandle (SQLSMALLINT    HandleType,
                          SQLHANDLE       InputHandle,
                          SQLHANDLE       *OutputHandle,
                          SQLSMALLINT     UseMultiThread);
```

- 파라미터

파라미터	용도	설명
HandleType	입력	할당 받을 핸들의 타입 – <code>SQL_HANDLE_ENV</code> – <code>SQL_HANDLE_DBC</code> – <code>SQL_HANDLE_STMT</code>
InputHandle	입력	입력 핸들(핸들의 타입에 따라 다른 값을 갖는다.) – <code>SQL_HANDLE_ENV</code> : <code>SQL_NULL_HANDLE</code> – <code>SQL_HANDLE_DBC</code> : 환경 핸들

파라미터	용도	설명
		- SQL_HANDLE_STMT: 연결 핸들
OutputHandle	출력	할당 받을 핸들의 포인터 (환경 핸들이 아닌 다른 핸들을 할당할 때 SQL_ERROR 를 반환하는 경우 OutputHandle 파라미터를 SQL_NULL_HDBC, SQL_NULL_HSTMT로 할당한다.)
UseMultiThread	입력	멀티 스레드의 사용 여부 (멀티 스레드 환경에서 tbCLI를 사용하는 경우 이 값을 SQL_TRUE로 설정한다.)

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
08003	데이터베이스 연결이 존재하지 않는 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY009	NULL 포인터를 잘못 사용한 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY014	한정된 범위의 핸들 개수를 초과한 경우
HY092	잘못된 속성 및 옵션 식별자
HYC00	선택적인 기능이 구현되지 않은 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLAllocHandle2 함수를 지원하지 않는 경우

- 관련 함수

이 함수는 SQLAllocConnect, SQLAllocEnv2, SQLAllocstmt를 대체하는 함수이다.

3.2.3. SQLGetWarningMsg

특정 수행문에 대해 경고 메시지가 발생할 경우, 해당 경고 메시지를 검색하는 함수이다.

SQLGetWarningMsg 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLGetWarningMsg (SQLHSTMT      StatementHandle,  
                             SQLCHAR        *OutputString,  
                             SQLINTEGER     *BufLength);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
OutputString	출력	반환될 경고 메시지를 담을 버퍼
BufLength	입력	반환될 경고 메시지를 담을 버퍼의 길이

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_ERROR	치명적인 에러가 발생한 상태

3.2.4. SQLGetWarningMsgLength

특정 수행문에 대해 경고 메시지가 발생할 경우, 해당 경고 메시지의 길이를 검색하는 함수이다.

SQLGetWarningMsgLength 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLGetWarningMsgLength (SQLHSTMT      StatementHandle,  
                                  SQLINTEGER     *Length);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
Length	출력	반환될 경고 메시지의 길이

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_ERROR	치명적인 에러가 발생한 상태

3.2.5. SQLLobClose

연산의 대상이 되는 LOB를 닫는 함수이다. 트랜잭션이 완료되기 전에 대상이 되는 LOB가 반드시 닫혀야 에러가 발생하지 않는다. 또한 열지 않은 LOB를 닫으려 할 때에도 에러가 발생한다.

그리고 Open 및 Close를 호출하지 않고 각 LOB와 관련된 함수를 호출하면 함수마다 내부적으로 Open 및 Close를 수행한다.

SQLLobClose 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLLobClose (SQLHSTMT      StatementHandle,
                        SQLINTEGER      Locator,
                        SQLINTEGER      *Indicator);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
Locator	입력	LOB locator
Indicator	입력	NULL 지시자(indicator)의 포인터

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
HY000	일반적인 에러
HY001	메모리 할당 에러
HY010	함수 시퀀스 에러

SQLSTATE	설명
HY013	메모리 관리 에러
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLLobClose 함수를 지원하지 않는 경우

3.2.6. SQLLobFreeLoc

연산의 대상이 되는 LOB Locator에 달려있는 자원을 해제한다.

SQLLobFreeLoc 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLLobFreeLoc (SQLHSTMT      StatementHandle,
                          SQLINTEGER     Locator);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
Locator	입력	LOB locator

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
HY000	일반적인 에러
HY001	메모리 할당 에러
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLLobFreeLoc 함수를 지원하지 않는 경우

3.2.7. SQLLobGetData(Deprecated)

원하는 위치의 대용량 객체형의 데이터를 가져오는 함수이다. 단, BLOB와 CLOB에 따라 가져오는 데이터가 다르다.

SQLLobGetData 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLLobGetData (SQLHSTMT      StatementHandle,
                          SQLINTEGER     SourceLocator,
                          SQLBIGINT      Offset,
                          SQLINTEGER     *ByteLength,
                          SQLINTEGER     *CharLength,
                          SQLSMALLINT    TargetCType,
                          SQLPOINTER     Data,
                          SQLINTEGER     BufferLength,
                          SQLINTEGER     *Indicator);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
SourceLocator	입력	LOB locator
Offset	입력	가져오려는 LOB 데이터의 시작 오프셋(offset) – CLOB/NCLOB: 시작 character의 위치 – BLOB: 시작 byte의 위치
ByteLength	입/출력	BLOB 타입의 요청 및 반환될 데이터의 길이 (bytes)
CharLength	입/출력	CLOB 타입의 요청 및 반환될 데이터의 길이 (characters)
TargetCType	입력	반환될 데이터의 C 타입
Data	입력	반환될 데이터 포인터
BufferLength	입력	반환될 데이터 버퍼의 길이
Indicator	출력	데이터에 쓰여진 byte 길이

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_NO_DATA	함수가 성공적으로 완료되었으나, 관련된 데이터를 찾을 수 없는 상태

반환 코드	설명
	(예: 주어진 SQL 문장에 의해 갱신된 로우가 없는 경우)
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_STILL_EXECUTING	이전에 실행한 SQL 문장이 완료되지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
01004	문자열 데이터가 오른쪽에서 잘린 경우
01S07	숫자 타입이나 날짜 타입의 일부 정보가 잘린 경우
07006	제한된 데이터 타입의 속성에 위배되는 경우
07009	잘못된 서술자 인덱스
08S01	서버와의 통신 연결에 실패한 경우
22002	지시자 변수가 요구되었으나 제공되지 않은 경우
22003	한정된 범위를 초과한 숫자 값인 경우
22007	잘못된 DATETIME 타입의 포맷인 경우
22012	0으로 나눈 경우
22015	INTERVAL 필드에 오버플로우가 발생한 경우
22018	캐스트 설정을 위한 문자 값이 잘못된 경우
24000	잘못된 커서 상태인 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY003	잘못된 애플리케이션의 버퍼 타입
HY008	연산이 취소된 경우
HY009	NULL 포인터를 잘못 사용한 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY090	잘못된 문자열 또는 버퍼 길이
HY109	잘못된 커서 위치
HYC00	선택적인 기능이 구현되지 않은 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLLobGetData 함수를 지원하지 않는 경우

- 관련 함수

SQLLobGetData 함수는 [SQLLobGetData2](#) 함수로 대체된다.

3.2.8. SQLLobGetData2

원하는 위치의 대용량 객체형의 데이터를 가져오는 함수이다. 단, BLOB과 CLOB에 따라 가져오는 데이터가 다르다.

SQLLobGetData2 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLLobGetData2 (SQLHSTMT      StatementHandle,
                           SQLINTEGER     Locator,
                           SQLBIGINT      Offset,
                           SQLINTEGER     *ReadLength,
                           SQLSMALLINT     TargetCType,
                           SQLPOINTER     DataPtr,
                           SQLINTEGER     BufferLength,
                           SQLINTEGER     *Indicator);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
Locator	입력	LOB locator
Offset	입력	가져오려는 LOB 데이터의 시작 오프셋(offset) – CLOB/NCLOB: 시작 character의 위치 – BLOB: 시작 byte의 위치
ReadLength	입/출력	– 입력 • CLOB/NCLOB: 가져올 character 길이 • BLOB: 가져올 byte 길이 – 출력 • CLOB/NCLOB: 가져온 character 길이 • BLOB: 가져온 byte 길이 0이면 데이터 버퍼 길이(BufferLength)만큼 다 채워서 반환

파라미터	용도	설명
TargetCType	입력	반환될 데이터의 C 타입
DataPtr	입력	반환될 데이터 포인터
BufferLength	입력	반환될 데이터 버퍼의 길이
Indicator	출력	데이터에 쓰여진 byte 길이

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_NO_DATA	함수가 성공적으로 완료되었으나, 관련된 데이터를 찾을 수 없는 상태 (예: 주어진 SQL 문장에 의해 갱신된 로우가 없는 경우)
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_STILL_EXECUTING	이전에 실행한 SQL 문장이 완료되지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
01004	문자열 데이터가 오른쪽에서 잘린 경우
01S07	숫자 타입이나 날짜 타입의 일부 정보가 잘린 경우
07006	제한된 데이터 타입의 속성에 위배되는 경우
07009	잘못된 서술자 인덱스
08S01	서버와의 통신 연결에 실패한 경우
22002	지시자 변수가 요구되었으나 제공되지 않은 경우
22003	한정된 범위를 초과한 숫자 값인 경우
22007	잘못된 DATETIME 타입의 포맷인 경우
22012	0으로 나눈 경우
22015	INTERVAL 필드에 오버플로우가 발생한 경우
22018	캐스트 설정을 위한 문자 값이 잘못된 경우
24000	잘못된 커서 상태인 경우
HY000	일반적인 에러
HY001	메모리 할당 에러

SQLSTATE	설명
HY003	잘못된 애플리케이션의 버퍼 타입
HY008	연산이 취소된 경우
HY009	NULL 포인터를 잘못 사용한 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY090	잘못된 문자열 또는 버퍼 길이
HY109	잘못된 커서 위치
HYC00	선택적인 기능이 구현되지 않은 경우
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLLobGetData 함수를 지원하지 않는 경우

3.2.9. SQLLobGetLength

특정 LOB 데이터의 길이를 검색하는 함수이다. BLOB와 CLOB 데이터 타입 모두에 사용할 수 있다.

SQLLobGetLength 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLLobGetLength (SQLHSTMT      StatementHandle,
                           SQLINTEGER      Locator,
                           SQLBIGINT      *StringLength,
                           SQLINTEGER      *Indicator);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
Locator	입력	LOB locator
StringLength	출력	반환될 LOB 데이터의 길이 – CLOB/NCLOB: character 길이 – BLOB: byte 길이
Indicator	입력	NULL 지시자의 포인터

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
07006	제한된 데이터 타입의 속성에 위배되는 경우
08S01	서버와의 통신 연결에 실패한 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY008	연산이 취소된 경우
HY009	NULL 포인터를 잘못 사용한 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY019	문자 및 바이너리가 아닌 데이터를 나누어 보낸 경우
HY020	NULL 값을 접합하려고 시도하는 경우
HY090	잘못된 문자열 또는 버퍼 길이
HYC00	선택적인 기능이 구현되지 않은 경우
0F001	LOB의 토큰에 아무런 값도 없는 경우

3.2.10. SQLLobOpen

연산의 대상이 될 LOB 데이터를 여는 함수이다. 이 함수는 각종 LOB 연산을 할 때 항상 먼저 선언되어야 한다. 그리고 Open 및 Close를 호출하지 않고 각 LOB와 관련된 함수를 호출하면 함수마다 내부적으로 Open 및 Close를 수행한다. 한 트랜잭션에서 같은 LOB 인스턴스를 두 번 열면 에러가 발생한다. 또한 다른 locator를 사용해 다시 열려고 해도 마찬가지로 에러가 발생한다.

SQLLobOpen 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLLobOpen (SQLHSTMT      StatementHandle,
                      SQLINTEGER      Locator,
```

```
SQLSMALLINT    Mode,
SQLINTEGER     *Indicator);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
SourceLocator	입력	LOB locator
Mode	입력	반환할 데이터의 위치 값
Indicator	입력	NULL 지시자의 포인터

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_STILL_EXECUTING	이전에 실행한 SQL 문장이 완료되지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
24000	잘못된 커서 상태인 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLLobOpen 함수를 지원하지 않는 경우

3.2.11. SQLLobPutData

LOB 내에서 원하는 위치에 데이터를 쓰는 함수이다.

SQLLobPutData 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLLobPutData (SQLHSTMT      StatementHandle,
                        SQLINTEGER      *DestLocator,
                        SQLBIGINT       DestOffset,
                        SQLSMALLINT     SourceCType,
                        SQLPOINTER      Source,
                        SQLINTEGER      *ByteLength,
                        SQLINTEGER      *CharLength,
                        SQLINTEGER      *Indicator);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
DestLocator	입력	LOB locator
DestOffset	입력	LOB 데이터에 쓸 위치
SourceCType	입력	쓸 데이터의 C의 타입
Source	입력	쓸 데이터의 포인터
ByteLength	출력	BLOB 타입의 쓴 데이터 길이
CharLength	출력	CLOB 타입의 쓴 데이터 길이
Indicator	입력	NULL 지시자의 포인터

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_STILL_EXECUTING	이전에 실행한 SQL 문장이 완료되지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
24000	잘못된 커서 상태인 경우
HY000	일반적 에러
HY001	메모리 할당 에러
HY010	함수 시퀀스 에러

SQLSTATE	설명
HY013	메모리 관리 에러
HYT01	데이터베이스 연결 제한 시간을 초과한 경우
IM001	드라이버가 SQLLobPutData 함수를 지원하지 않는 경우

3.2.12. SQLLobTruncate

LOB 데이터를 잘라내는 함수이다.

SQLLobTruncate 함수의 세부 내용은 다음과 같다.

- 문법

```
SQLRETURN SQLLobTruncate (SQLHSTMT      StatementHandle,
                           SQLINTEGER     *Locator,
                           SQLBIGINT      NewLength,
                           SQLINTEGER     *Indicator);
```

- 파라미터

파라미터	용도	설명
StatementHandle	입력	문장 핸들
Locator	입력	LOB locator
NewLength	입력	잘린 후 LOB 데이터의 새로운 길이
Indicator	입력	NULL 지시자의 포인터

- 반환 코드

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_STILL_EXECUTING	이전에 실행한 SQL 문장이 완료되지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

- SQLSTATE

SQLSTATE	설명
01000	일반적인 경고
01004	문자열 데이터가 오른쪽에서 잘린 경우

SQLSTATE	설명
07006	제한된 데이터 타입의 속성에 위배되는 경우
07S01	디폴트 파라미터가 잘못 사용된 경우
08S01	서버와의 통신 연결에 실패한 경우
HY000	일반적인 에러
HY001	메모리 할당 에러
HY008	연산이 취소된 경우
HY009	NULL 포인터를 잘못 사용한 경우
HY010	함수 시퀀스 에러
HY013	메모리 관리 에러
HY019	문자 및 바이너리가 아닌 데이터를 나누어 보낸 경우
HY090	잘못된 문자열 또는 버퍼 길이
HYT01	데이터베이스 연결 제한 시간을 초과한 경우

제4장 tbCLI 에러 메시지

본 장에서는 tbCLI 프로그램을 실행하는 과정에서 발생할 수 있는 에러를 설명한다.

4.1. 반환 코드

모든 tbCLI 함수는 실행 후 반환 코드를 반환한다. 반환 코드는 **SQLRETURN** 타입이며 미리 정해진 값 중의 하나이다.

tbCLI 함수는 다음 표와 같이 반환되는 반환 코드에 따라 tbCLI 프로그램의 상태를 확인할 수 있다.

반환 코드	설명
SQL_SUCCESS	함수가 성공적으로 완료된 상태
SQL_SUCCESS_WITH_INFO	함수가 성공적으로 완료되었으나, 경고 메시지가 있는 상태
SQL_NO_DATA	함수가 성공적으로 완료되었으나, 관련된 데이터를 찾을 수 없는 상태 (예: 주어진 SQL 문장에 의해 갱신된 로우가 없는 경우)
SQL_INVALID_HANDLE	입력 파라미터에 주어진 핸들이 유효하지 않은 상태
SQL_NEED_DATA	SQL 문장을 실행하기 위해 데이터가 더 필요한 상태 (예: 파라미터)
SQL_STILL_EXECUTING	이전에 실행한 SQL 문장이 완료되지 않은 상태
SQL_ERROR	치명적인 에러가 발생한 상태

이러한 반환 코드는 추가적으로 SQLSTATE 정보가 존재할 수 있다.

참고

SQLSTATE에 대한 내용은 관련 API나 ["4.2. SQLSTATE"](#)을 참고한다.

4.2. SQLSTATE

본 절에서는 SQLSTATE 정보를 코드 순으로 나열하고 이를 설명한다. 추가로 해당 SQLSTATE 코드를 반환할 수 있는 함수도 포함하여 기술한다.

4.2.1. 01000 ~01007

- 01000

설명	일반적인 경고를 뜻하는 에러이다.
함수	SQLError, SQLGetDiagField, SQLGetDiagRec를 제외한 모든 함수

- 01001

설명	커서 연산이 충돌했을 때 발생하는 에러이다.
함수	SQLExecDirec, SQLExecute, SQLParamDat, SQLSetPos

- 01002

설명	데이터베이스 연결을 해제하는 중에 발생하는 에러이다.
함수	SQLError, SQLGetDiagField, SQLGetDiagRec

- 01003

설명	설정된 함수에서 NULL 값을 제거한 경우에 발생하는 에러이다.
함수	SQLExecDirect, SQLExecute, SQLParamData

- 01004

설명	문자열 데이터가 오른쪽에서 잘릴 때 발생하는 에러이다.
함수	SQLBrowseConnect, SQLBulkOperations, SQLColAttribute, SQLDataSources, SQLDescribeCol, SQLDriverConnect, SQLDrivers, SQLExecDirect, SQLExecute, SQLExtendedFetch, SQLFetch, SQLFetchScroll, SQLGetConnectAttr, SQLGetCursorName, SQLGetData, SQLGetDescField, SQLGetDescRec, SQLGetEnvAttr, SQLGetInfo, SQLGetStmtAttr, SQLNativeSql, SQLParamData, SQLPutData, SQLSetCursorName

- 01006

설명	취소되지 않은 특권일 때 발생하는 에러이다.
함수	SQLExecDirec, SQLExecute, SQLParamData

- 01007

설명	부여되지 않은 특권일 때 발생하는 에러이다.
함수	SQLExecDirect, SQLExecute, SQLParamData

4.2.2. 01S00 ~ 01S09

- 01S00

설명	유효하지 않은 연결 문자열의 속성일 때 발생하는 에러이다.
함수	SQLBrowseConnect, SQLDriverConnect

- 01S01

설명	로우에 에러가 있는 경우 발생한다.
함수	SQLBulkOperations, SQLExtendedFetch, SQLSetPos

- 01S02

설명	옵션 값이 변경되었을 때 발생하는 에러이다.
함수	다음의 함수를 제외한 모든 함수 SQLBrowseConnect, SQLConnect, SQLDriverConnect, SQLExecDirect, SQLExecute, SQLParamData, SQLPrepare, SQLSetConnectAttr, SQLSetDescField, SQLSetEnvAttr, SQLSetStmtAttr

- 01S06

설명	결과 집합의 첫 번째 로우가 반환되기 전에 페치를 시도한 경우 발생하는 에러이다.
함수	SQLExtendedFetch, SQLFetchScroll

- 01S07

설명	숫자 타입이나 날짜 타입의 일부 정보가 잘릴 때 발생하는 에러이다.
함수	SQLBulkOperations, SQLExecDirect, SQLExecute, SQLExtendedFetch, SQLFetch, SQLFetchScroll, SQLGetData, SQLParamData, SQLSetPos

- 01S08

설명	DSN 파일(.dsn)을 저장하지 못할 때 발생하는 에러이다.
함수	SQLDriverConnect

- 01S09

설명	잘못된 예약어일 때 발생하는 에러이다.
함수	SQLDriverConnect

4.2.3. 07001 ~ 07009

- 07001

설명	파라미터의 숫자가 잘못되었을 때 발생하는 에러이다.
함수	SQLExecDirect, SQLExecute

- 07002

설명	COUNT 필드에 오류가 있는 경우에 발생하는 에러이다.
함수	SQLExecDirect, SQLExecute, SQLParamData

- 07005

설명	커서의 범위가 아닌 준비된 문장인 경우에 발생하는 에러이다.
함수	SQLColAttribute, SQLDescribeCol

- 07006

설명	제한된 데이터 타입의 속성에 위배될 때 발생하는 에러이다.
함수	SQLBindCol, SQLBindParameter, SQLBulkOperations, SQLExecDirect, SQLExecute, SQLExtendedFetch, SQLFetch, SQLFetchScroll, SQLGetData, SQLParamData, SQLPutData, SQLSetPos

- 07009

설명	잘못된 서술자 인덱스일 때 발생하는 에러이다.
함수	SQLBindCol, SQLBindParameter, SQLBulkOperations, SQLColAttribute, SQLDescribeCol, SQLDescribeParam, SQLFetch, SQLFetchScroll, SQLGetData, SQLGetDescField, SQLGetDescRec, SQLParamData, SQLSetDescField, SQLSetDescRec, SQLSetPos

4.2.4. 07S01

- 07S01

설명	디폴트 파라미터를 잘못 사용한 경우 발생하는 에러이다.
함수	SQLExecDirect, SQLExecute, SQLParamData, SQLPutData

4.2.5. 08001 ~ 08007

- 08001

설명	클라이언트가 데이터소스에 연결할 수 없을 때 발생하는 에러이다.
함수	SQLBrowseConnect, SQLConnect, SQLDriverConnect

- 08002

설명	이미 사용 중인 데이터베이스 연결명일 때 발생하는 에러이다.
함수	SQLBrowseConnect, SQLConnect, SQLDriverConnect, SQLSetConnectAttr

- 08003

설명	연결이 존재하지 않을 때 발생하는 에러이다.
함수	SQLAllocHandle, SQLDisconnect, SQLEndTran, SQLGetConnectAttr, SQLGetInfo, SQLNativeSql, SQLSetConnectAttr

- 08004

설명	서버가 연결을 거부할 때 발생하는 에러이다.
함수	SQLBrowseConnect, SQLConnect, SQLDriverConnect

- 08007

설명	트랜잭션을 수행하는 중에 연결이 실패한 경우 발생하는 에러이다.
함수	SQLEndTran

4.2.6. 08S01

- 08S01

설명	통신 연결이 실패한 경우 발생하는 에러이다.
함수	SQLBrowseConnect, SQLColumnPrivileges, SQLColumns, SQLConnect, SQLCopyDesc, SQLDescribeCol, SQLDescribeParam, SQLDriverConnect, SQLExecDirect, SQLExecute, SQLExtendedFetch, SQLFetch, SQLFetchScroll, SQLForeignKeys, SQLGetConnectAttr, SQLGetData, SQLGetDescField, SQLGetDescRec, SQLGetFunctions, SQLGetInfo, SQLGetTypeInfo, SQLMoreResults, SQLNativeSql, SQLNumParams, SQLNumResultCols,

	SQLParamData, SQLPrepare, SQLPrimaryKeys, SQLProcedureColumns, SQLProcedures, SQLPutData, SQLSetConnectAttr, SQLSetDescField, SQLSetDescRec, SQLSetEnvAttr, SQLSetStmtAttr, SQLSpecialColumns, SQLStatistics, SQLTablePrivileges, SQLTables
--	---

4.2.7. 21S01 ~ 21S02

- 21S01

설명	삽입될 값의 리스트가 컬럼과 일치하지 않을 때 발생하는 에러이다.
함수	SQLExecDirect, SQLPrepare

- 21S02

설명	파생된 테이블의 degree가 컬럼과 일치하지 않을 때 발생하는 에러이다.
함수	SQLBulkOperations, SQLExecDirect, SQLExecute, SQLParamData, SQLPrepare, SQLSetPos

4.2.8. 22001 ~ 22026

- 22001

설명	문자열 데이터가 오른쪽에서 잘릴 때 발생하는 에러이다.
함수	SQLBulkOperations, SQLExecDirect, SQLExecute, SQLFetch, SQLFetchScroll, SQLParamData, SQLPutData, SQLSetDescField, SQLSetPos

- 22002

설명	지시자 변수가 요구되었으나, 제공되지 않았을 때 발생하는 에러이다.
함수	SQLExecDirect, SQLExecute, SQLExtendedFetch, SQLFetch, SQLFetchScroll, SQLGetData, SQLParamData

- 22003

설명	한정된 범위를 초과한 숫자 값일 때 발생하는 에러이다.
함수	SQLBulkOperations, SQLExecDirect, SQLExecute, SQLExtendedFetch, SQLFetch,

	SQLFetchScroll, SQLGetData, SQLGetInfo, SQLParamData, SQLPutData, SQLSetPos
--	---

- 22007

설명	유효하지 않은 날짜형(DATETIME) 형식일 때 발생하는 에러이다.
함수	SQLBulkOperations, SQLExecDirect, SQLExecute, SQLExtendedFetch, SQLFetch, SQLFetchScroll, SQLGetData, SQLParamData, SQLPutData, SQLSetPos

- 22008

설명	날짜형 필드에 오버플로우가 발생했을 때 나타나는 에러이다.
함수	SQLBulkOperations, SQLExecDirect, SQLExecute, SQLParamData, SQLPutData

- 22012

설명	0으로 나누었을 때 발생하는 에러이다.
함수	SQLExecDirect, SQLExecute, SQLExtendedFetch, SQLFetch, SQLFetchScroll, SQLGetData, SQLParamData, SQLPutData

- 22015

설명	INTERVAL 필드에 오버플로우가 발생했을 때 나타나는 에러이다.
함수	SQLBulkOperations, SQLExecDirect, SQLExecute, SQLExtendedFetch, SQLFetch, SQLFetchScroll, SQLGetData, SQLParamData, SQLPutData, SQLSetPos

- 22018

설명	캐스트 설정을 위한 문자 값이 잘못된 경우에 발생하는 에러이다.
함수	SQLBulkOperations, SQLExecDirect, SQLExecute, SQLExtendedFetch, SQLFetch, SQLFetchScroll, SQLGetData, SQLParamData, SQLPutData, SQLSetPos

- 22019

설명	잘못된 이스케이프 문자일 때 발생하는 에러이다.
함수	SQLExecDirect, SQLExecute, SQLPrepare

- 22025

설명	잘못된 이스케이프 시퀀스일 때 발생하는 에러이다.
-----------	-----------------------------

함수	SQLExecDirect, SQLExecute, SQLPrepare
-----------	---------------------------------------

- 22026

설명	잘못된 길이를 가진 문자열 데이터일 때 발생하는 에러이다.
함수	SQLParamData

4.2.9. 23000, 24000, 25000

- 23000

설명	무결성 제약조건에 위배될 때 발생하는 에러이다.
함수	SQLBulkOperations, SQLExecDirect, SQLExecute, SQLParamData, SQLSetPos

- 24000

설명	잘못된 커서 상태일 때 발생하는 에러이다.
함수	SQLBulkOperations, SQLCloseCursor, SQLColumnPrivileges, SQLColumns, SQLExecDirect, SQLExecute, SQLExtendedFetch, SQLFetch, SQLFetchScroll, SQLForeignKeys, SQLGetData, SQLGetStmtAttr, SQLGetTypeInfo, SQLNativeSql, SQLPrepare, SQLPrimaryKeys, SQLProcedureColumns, SQLProcedures, SQLSetConnectAttr, SQLSetCursorName, SQLSetPos, SQLSetStmtAttr, SQLSpecialColumns, SQLStatistics, SQLTablePrivileges, SQLTables

- 25000

설명	잘못된 트랜잭션 상태일 때 발생하는 에러이다.
함수	SQLDisconnect

4.2.10. 25S01 ~ 25S03

- 25S01

설명	트랜잭션 상태일 때 발생하는 에러이다.
함수	SQLEndTran

- 25S02

설명	트랜잭션이 아직 활성화 중일 때 발생하는 에러이다.
함수	SQLEndTran

- 25S03

설명	트랜잭션이 롤백되었을 때 발생하는 에러이다.
함수	SQLEndTran

4.2.11. 28000, 34000

- 28000

설명	잘못된 권한을 설정했을 때 발생하는 에러이다.
함수	SQLBrowseConnect, SQLConnect, SQLDriverConnect

- 34000

설명	잘못된 커서 이름일 때 발생하는 에러이다.
함수	SQLExecDirect, SQLPrepare, SQLSetCursorName

4.2.12. 3C000 ~ 3F000

- 3C000

설명	중복된 커서 이름일 때 발생하는 에러이다.
함수	SQLSetCursorName

- 3D000

설명	잘못된 카탈로그 이름일 때 발생하는 에러이다.
함수	SQLExecDirect, SQLPrepare, SQLSetConnectAttr

- 3F000

설명	잘못된 스키마 이름일 때 발생하는 에러이다.
함수	SQLExecDirect, SQLPrepare

4.2.13. 40001 ~ 40003, 42000

- 40001

설명	직렬화에 실패했을 때 발생하는 에러이다.
함수	SQLBulkOperations, SQLColumnPrivileges, SQLColumns, SQLEndTran, SQLExecDirect, SQLExecute, SQLFetch, SQLFetchScroll, SQLForeignKeys, SQLGetTypeInfo, SQLMoreResults, SQLParamData, SQLPrimaryKeys, SQLProcedureColumns, SQLProcedures, SQLSetPos, SQLSpecialColumns, SQLStatistics, SQLTablePrivileges, SQLTables

- 40002

설명	무결성 제약조건에 위배될 때 발생하는 에러이다.
함수	SQLEndTran

- 40003

설명	실행이 완료되지 않았고, 트랜잭션의 상태를 알 수 없을 때 발생하는 에러이다.
함수	SQLBulkOperations, SQLColumnPrivileges, SQLColumns, SQLExecDirect, SQLExecute, SQLFetch, SQLFetchScroll, SQLForeignKeys, SQLGetTypeInfo, SQLMoreResults, SQLPrimaryKeys, SQLProcedureColumns, SQLProcedures, SQLParamData, SQLSetPos, SQLSpecialColumns, SQLStatistics, SQLTablePrivileges, SQLTables

- 42000

설명	문법 에러 또는 접근 위배가 발생한 경우 나타나는 에러이다.
함수	SQLBulkOperations, SQLExecDirect, SQLExecute, SQLParamData, SQLPrepare, SQLSetPos

4.2.14. 42S01 ~ 42S22, 44000

- 42S01

설명	기본 테이블 또는 뷰가 이미 존재할 때 발생하는 에러이다.
함수	SQLExecDirect, SQLPrepare

- 42S02

설명	기반 테이블 또는 뷰를 찾을 수 없을 때 발생하는 에러이다.
함수	SQLExecDirect, SQLPrepare

- 42S11

설명	인덱스가 이미 존재할 때 발생하는 에러이다.
함수	SQLExecDirect, SQLPrepare

- 42S12

설명	인덱스를 찾을 수 없을 때 발생하는 에러이다.
함수	SQLExecDirect, SQLPrepare

- 42S21

설명	컬럼이 이미 존재할 때 발생하는 에러이다.
함수	SQLExecDirect, SQLPrepare

- 42S22

설명	컬럼을 찾을 수 없을 때 발생하는 에러이다.
함수	SQLExecDirect, SQLPrepare

- 44000

설명	WITH CHECK OPTION에 위배될 때 발생하는 에러이다.
함수	SQLBulkOperations, SQLExecDirect, SQLExecute, SQLParamData, SQLSetPos

4.2.15. HY000 ~ HY024

- HY000

설명	일반적인 에러를 뜻하는 에러이다.
함수	(All ODBC functions except) SQLError, SQLGetDiagField, SQLGetDiagRec

- HY001

설명	메모리 할당에 실패했을 때 발생하는 에러이다.
함수	(All ODBC functions except) SQLError, SQLGetDiagField, SQLGetDiagRec

- HY003

설명	잘못된 애플리케이션의 버퍼 타입일 때 발생하는 에러이다.
함수	SQLBindCol, SQLBindParameter, SQLGetData

- HY004

설명	잘못된 SQL의 데이터 타입일 때 발생하는 에러이다.
함수	SQLBindParameter, SQLGetTypeInfo

- HY007

설명	해당 문장이 준비되지 않았을 때 발생하는 에러이다.
함수	SQLCopyDesc, SQLGetDescField, SQLGetDescRec

- HY008

설명	연산이 취소되었을 때 발생하는 에러이다.
함수	SQLBulkOperations, SQLColAttribute, SQLColumnPrivileges, SQLColumns, SQLDescribeCol, SQLDescribeParam, SQLExecDirect, SQLExecute, SQLExtendedFetch, SQLFetch, SQLFetchScroll, SQLForeignKeys, SQLGetData, SQLGetTypeInfo, SQLMoreResults, SQLNumParams, SQLNumResultCols, SQLParamData, SQLPrepare, SQLPrimaryKeys, SQLProcedureColumns, SQLProcedures, SQLPutData, SQLSetPos, SQLSpecialColumns, SQLStatistics, SQLTablePrivileges, SQLTables

- HY009

설명	NULL 포인터를 잘못 사용했을 때 발생하는 에러이다.
함수	SQLAllocHandle, SQLBindParameter, SQLBulkOperations, SQLColumnPrivileges, SQLColumns, SQLExecDirect, SQLForeignKeys, SQLGetCursorName, SQLGetData, SQLGetFunctions, SQLNativeSql, SQLPrepare, SQLPrimaryKeys, SQLProcedureColumns, SQLProcedures, SQLPutData, SQLSetConnectAttr, SQLSetCursorName, SQLSetEnvAttr, SQLSetStmtAttr, SQLSpecialColumns, SQLStatistics, SQLTablePrivileges, SQLTables

- HY010

설명	함수 시퀀스에 문제가 있는 경우 발생하는 에러이다.
함수	SQLAllocHandle, SQLBindCol, SQLBindParameter, SQLBulkOperations, SQLCloseCursor,

	SQLColAttribute, SQLColumnPrivileges, SQLColumns, SQLCopyDesc, SQLDescribeCol, SQLDescribeParam, SQLDisconnect, SQLEndTran, SQLExecDirect, SQLExecute, SQLExtendedFetch, SQLFetch, SQLFetchScroll, SQLForeignKeys, SQLFreeHandle, SQLFreeStmt, SQLGetConnectAttr, SQLGetCursorName, SQLGetData, SQLGetDescField, SQLGetDescRec, SQLGetFunctions, SQLGetStmtAttr, SQLGetTypeInfo, SQLMoreResults, SQLNumParams, SQLNumResultCols, SQLParamData, SQLPrepare, SQLPrimaryKeys, SQLProcedureColumns, SQLProcedures, SQLPutData, SQLRowCount, SQLSetConnectAttr, SQLSetCursorName, SQLSetDescField, SQLSetEnvAttr, SQLSetDescRec, SQLSetPos, SQLSetStmtAttr, SQLSpecialColumns, SQLStatistics, SQLTablePrivileges, SQLTables
--	--

- HY011

설명	현재 설정될 수 없는 속성일 때 발생하는 에러이다.
함수	SQLBulkOperations, SQLParamData, SQLSetConnectAttr, SQLSetPos, SQLSetStmtAttr

- HY012

설명	잘못된 트랜잭션의 연산 코드일 때 발생하는 에러이다.
함수	SQLEndTran

- HY013

설명	메모리 관리에 에러가 발생한 경우 나타나는 에러이다.
함수	(All ODBC functions except) SQLGetDiagField, SQLGetDiagRec

- HY014

설명	한정된 핸들 개수를 초과했을 때 발생하는 에러이다.
함수	SQLAllocHandle

- HY015

설명	사용할 수 있는 커서 이름이 없을 때 발생하는 에러이다.
함수	SQLGetCursorName

- HY016

설명	구현 로우 서술자를 수정할 수 없을 때 발생하는 에러이다.
-----------	----------------------------------

함수	SQLCopyDesc, SQLSetDescField, SQLSetDescRec
-----------	---

- HY017

설명	자동으로 할당된 서술자 핸들을 잘못 사용했을 때 발생하는 에러이다.
함수	SQLFreeHandle, SQLSetStmtAttr

- HY018

설명	서버가 취소 요구를 거부했을 때 발생하는 에러이다.
함수	SQLCancel

- HY019

설명	비문자열(Non-character)과 비이진 데이터(Non-binary data)를 여러 조각으로 나누어 보냈을 때 발생하는 에러이다.
함수	SQLPutData

- HY020

설명	NULL 값을 접합하려고 시도했을 때 발생하는 에러이다.
함수	SQLPutData

- HY021

설명	모순이 있는 서술자 정보일 때 발생하는 에러이다.
함수	SQLBindParameter, SQLCopyDesc, SQLGetDescField, SQLSetDescField, SQLSetDescRec

- HY024

설명	잘못된 속성 값일 때 발생하는 에러이다.
함수	SQLSetConnectAttr, SQLSetEnvAttr, SQLSetStmtAttr

4.2.16. HY090 ~ HY111

- HY090

설명	잘못된 문자열 또는 버퍼 길이일 때 발생하는 에러이다.
함수	SQLBindCol, SQLBindParameter, SQLBrowseConnect, SQLBulkOperations, SQLColAttribute, SQLColumnPrivileges, SQLColumns, SQLConnect, SQLDataSources, SQLDescribeCol, SQLDriverConnect, SQLDrivers, SQLExecDirect, SQLExecute,

	SQLFetch, SQLFetchScroll, SQLForeignKeys, SQLGetConnectAttr, SQLGetCursorName, SQLGetData, SQLGetDescField, SQLGetInfo, SQLGetStmtAttr, SQLNativeSql, SQLParamData, SQLPrepare, SQLPrimaryKeys, SQLProcedureColumns, SQLProcedures, SQLPutData, SQLSetConnectAttr, SQLSetCursorName, SQLSetDescField, SQLSetDescRec, SQLSetEnvAttr, SQLSetStmtAttr, SQLSetPos, SQLSpecialColumns, SQLStatistics, SQLTablePrivileges, SQLTables
--	--

- HY091

설명	잘못된 서술자의 필드 식별자일 때 발생하는 에러이다.
함수	SQLColAttribute, SQLGetDescField, SQLSetDescField

- HY092

설명	잘못된 속성 및 옵션 식별자일 때 발생하는 에러이다.
함수	SQLAllocHandle, SQLBulkOperations, SQLCopyDesc, SQLDriverConnect, SQLEndTran, SQLFreeStmt, SQLGetConnectAttr, SQLGetEnvAttr, SQLGetStmtAttr, SQLParamData, SQLSetConnectAttr, SQLSetDescField, SQLSetEnvAttr, SQLSetPos, SQLSetStmtAttr

- HY095

설명	영역 밖의 함수 타입인 경우 발생하는 에러이다.
함수	SQLGetFunctions

- HY096

설명	잘못된 정보 타입인 경우 발생하는 에러이다.
함수	SQLGetInfo

- HY097

설명	영역 밖의 컬럼인 경우 발생하는 에러이다.
함수	SQLSpecialColumns

- HY098

설명	영역 밖의 범위 타입인 경우 발생하는 에러이다.
함수	SQLSpecialColumns

- HY099

설명	영역 밖의 Nullable 타입인 경우 발생하는 에러이다.
함수	SQLSpecialColumns

- HY100

설명	영역 밖의 고유 옵션 타입인 경우 발생하는 에러이다.
함수	SQLStatistics

- HY101

설명	영역 밖의 정확도 옵션 타입인 경우 발생하는 에러이다.
함수	SQLStatistics

- HY103

설명	잘못된 검색(retrieval) 코드인 경우 발생하는 에러이다.
함수	SQLDataSources, SQLDrivers

- HY104

설명	잘못된 정밀도 또는 스케일 값인 경우 발생하는 에러이다.
함수	SQLBindParameter

- HY105

설명	잘못된 파라미터 타입인 경우 발생하는 에러이다.
함수	SQLBindParameter, SQLExecDirect, SQLExecute, SQLParamData, SQLSetDescField

- HY106

설명	영역 밖의 페치 타입인 경우 발생하는 에러이다.
함수	SQLExtendedFetch, SQLFetchScroll

- HY107

설명	영역 밖의 로우 값인 경우 발생하는 에러이다.
함수	SQLExtendedFetch, SQLFetch, SQLFetchScroll, SQLSetPos

- HY109

설명	잘못된 커서 위치일 때 발생하는 에러이다.
함수	SQLExecDirect, SQLExecute, SQLGetData, SQLGetStmtAttr, SQLNativeSql,

	SQLParamData, SQLSetPos
--	-------------------------

- HY110

설명	잘못된 드라이버 완성(driver completion)일 때 발생하는 에러이다.
함수	SQLDriverConnect

- HY111

설명	잘못된 북마크 값인 경우 발생하는 에러이다.
함수	SQLExtendedFetch, SQLFetchScroll

4.2.17. HYC00

- HYC00

설명	구현되지 않은 선택적인 기능(Optional feature)을 사용하려 할 때 발생하는 에러이다.
함수	SQLBindCol, SQLBindParameter, SQLBulkOperations, SQLColAttribute, SQLColumnPrivileges, SQLColumns, SQLDriverConnect, SQLEndTran, SQLExecDirect, SQLExecute, SQLExtendedFetch, SQLFetch, SQLFetchScroll, SQLForeignKeys, SQLGetConnectAttr, SQLGetData, SQLGetEnvAttr, SQLGetInfo, SQLGetStmtAttr, SQLGetTypeInfo, SQLParamData, SQLPrepare, SQLPrimaryKeys, SQLProcedureColumns, SQLProcedures, SQLSetConnectAttr, SQLSetEnvAttr, SQLSetPos, SQLSetStmtAttr, SQLSpecialColumns, SQLStatistics, SQLTablePrivileges, SQLTables

4.2.18. HYT00 ~ HYT01

- HYT00

설명	제한 시간을 초과했을 때 발생하는 에러이다.
함수	SQLBrowseConnect, SQLBulkOperations, SQLColumnPrivileges, SQLColumns, SQLConnect, SQLDriverConnect, SQLExecDirect, SQLExecute, SQLExtendedFetch, SQLForeignKeys, SQLGetTypeInfo, SQLParamData, SQLPrepare, SQLPrimaryKeys, SQLProcedureColumns, SQLProcedures, SQLSetPos, SQLSpecialColumns,

	SQLStatistics, SQLTablePrivileges, SQLTables
--	--

- HYT01

설명	데이터베이스 연결 제한 시간을 초과했을 때 발생하는 에러이다.
함수	(All ODBC functions except) SQLDrivers, SQLDataSources, SQLGetEnvAttr, SQLSetEnvAttr

4.2.19. IM001 ~ IM015

- IM001

설명	드라이버가 해당 함수를 제공하지 않을 때 발생하는 에러이다.
함수	(All ODBC functions except) SQLAllocHandle, SQLDataSources, SQLDrivers, SQLFreeHandle, SQLGetFunctions

- IM002

설명	데이터소스의 이름을 찾을 수 없거나, 디폴트 드라이버가 존재하지 않을 때 발생하는 에러이다.
함수	SQLBrowseConnect, SQLConnect, SQLDriverConnect

- IM003

설명	지정한 드라이버를 로드할 수 없을 때 발생하는 에러이다.
함수	SQLBrowseConnect, SQLConnect, SQLDriverConnect

- IM004

설명	해당 드라이버에 SQLAllocHandle 함수를 이용한 환경 핸들 할당에 실패했을 때 발생하는 에러이다.
함수	SQLBrowseConnect, SQLConnect, SQLDriverConnect

- IM005

설명	해당 드라이버에 SQLAllocHandle 함수를 이용한 연결 핸들 할당에 실패했을 때 발생하는 에러이다.
함수	SQLBrowseConnect, SQLConnect, SQLDriverConnect

- IM006

설명	해당 드라이버에 SQLSetConnectAttr 함수를 이용한 데이터베이스 연결 속성 설정에 실패했을 때 발생하는 에러이다.
함수	SQLBrowseConnect, SQLConnect, SQLDriverConnect

- IM007

설명	데이터소스 또는 드라이버가 지정되지 않았을 때 발생하는 에러이다.
함수	SQLDriverConnect

- IM009

설명	Translation DLL를 로드하는 것이 불가능할 때 발생하는 에러이다.
함수	SQLBrowseConnect, SQLConnect, SQLDriverConnect, SQLSetConnectAttr

- IM010

설명	데이터소스의 이름이 SQL_MAX_DSN_LENGTH 파라미터에 설정된 값보다 길 때 발생하는 에러이다.
함수	SQLBrowseConnect, SQLConnect, SQLDriverConnect

- IM011

설명	드라이버 이름이 길 때 발생하는 에러이다.
함수	SQLBrowseConnect, SQLDriverConnect

- IM012

설명	드라이버 예약어에 문법 에러가 발생했을 때 나타나는 에러이다.
함수	SQLBrowseConnect, SQLDriverConnect

- IM013

설명	트레이스 파일에 문제가 발생한 경우 나타나는 에러이다.
함수	All ODBC functions

- IM014

설명	잘못된 파일 DSN의 이름일 때 발생하는 에러이다.
함수	SQLDriverConnect

- IM015

설명	손상된 파일의 데이터소스일 때 발생하는 에러이다.
함수	SQLDriverConnect

4.2.20. 00000, 02000

- 00000

설명	함수가 성공적으로 완료된 상태이다.
함수	모든 함수

- 02000

설명	함수가 성공적으로 완료되었으나, 관련된 데이터를 찾을 수 없는 상태이다.
함수	모든 함수

제5장 tbCLI와 ODBC

본 장에서는 tbCLI와 ODBC를 연동하는 방법을 설명한다.

5.1. ODBC

ODBC는 아래와 같은 Call-Level Interface(CLI) 명세와 표준을 따르는 데이터베이스의 API이다.

The Open Group CAE Specification "Data Management: SQL Call-Level Interface (CLI)"
ISO/IEC 9075-3:1995 (E) Call-Level Interface (SQL/CLI)

ODBC를 사용하게 되면 동일한 애플리케이션 프로그램으로 다양한 벤더의 DBMS에 접근할 수 있다. 이를 위해서는 ODBC 소프트웨어 외에 액세스할 데이터베이스마다 별도의 모듈이나 드라이버가 필요하다.

5.2. tbCLI와 ODBC의 연동

tbCLI는 Call-Level Interface(CLI) 명세와 표준을 따르므로, ODBC와 연동할 수 있다. tbCLI는 ODBC 3.51 표준에 맞게 구현되어 있어 데이터베이스 추적 로그 등 ODBC가 가진 기능을 사용할 수 있다.

Tibero에서는 ODBC와의 연동을 위해 운영체제별로 ODBC 드라이버를 제공한다.

5.2.1. Windows 계열에서의 설치

Windows 계열에서 **Tibero 6 ODBC Driver**를 설치할 때 사용자의 편의를 위해 Tibero에서는 Win32/Win64용 ODBC 드라이버 설치 실행 파일(Win32: **tbodbc_driver_installer_6_32.exe**, Win64: **tbodbc_driver_installer_6_64.exe**)과 Win32/Win64용 ODBC 클라이언트 라이브러리 파일(**libtbcli.dll**)을 제공하고 있다.

이 두 파일은 다음과 같은 디렉터리에 존재한다.

- Win32용 배포판의 경우

```
%TB_HOME%\bin\tbodbc_driver_installer_6_32.exe  
%TB_HOME%\bin\libtbcli.dll
```

- Win64용 배포판의 경우

```
%TB_HOME%\bin\tbodbc_driver_installer_6_64.exe  
%TB_HOME%\bin\libtbcli.dll
```

- UNIX용 배포판의 경우

```
$TB_HOME/client/win32/bin/tbodbc_driver_installer_6_32.exe
$TB_HOME/client/win32/lib/libtbcli.dll
$TB_HOME/client/win64/bin/tbodbc_driver_installer_6_64.exe
$TB_HOME/client/win64/lib/libtbcli.dll
```

설치

Win32용 드라이버 설치 순서는 다음과 같다.

1. Win32용 libtbcli.dll 파일을 설치할 폴더에 복사한다. ODBC 데이터 원본 관리자에서 libtbcli.dll 파일을 동적으로 링크하여 사용한다.
2. 명령 프롬프트(cmd.exe) 창을 실행한 후 다음과 같은 명령을 실행한다.

```
%TB_HOME%\bin\tbodbc_driver_installer_6_32.exe -i [설치 경로]
```

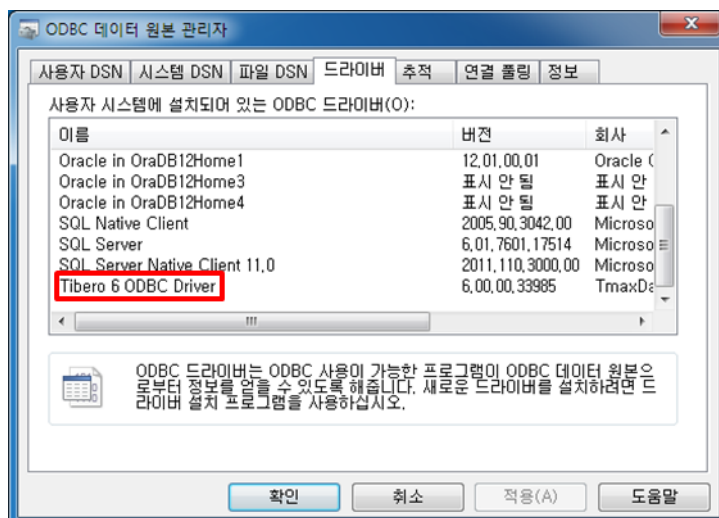
설치 경로는 복사한 libtbcli.dll 파일이 있는 디렉터리까지의 절대 경로를 입력한다. 이때 설치 경로를 입력하지 않으면 디폴트로 설정된 %TB_HOME%\bin 폴더에 설치된다.

설치 확인

설치 후 확인하는 순서는 다음과 같다.

1. **ODBC 데이터 원본 관리자**를 실행한다. 실행하는 방법은 **[제어판]>[관리 도구]>[데이터 원본 (ODBC)]** 메뉴를 선택하거나 또는 %SystemRoot%\system32 폴더에서 odbcad32.exe 파일을 실행하면 된다. 만약 64Bit 머신에서 Win32용 드라이버를 설치하였다면 %SystemRoot%\SysWOW64 폴더에서 odbcad32.exe 파일을 실행해야 한다.
2. **[드라이버]** 탭으로 이동한 후 **Tibero 6 ODBC Driver**가 있는지 확인한다.

[그림 5.1] Tibero 6 ODBC Driver 설치 확인



DSN 등록

Tibero 6 ODBC Driver를 정상적으로 설치했으면, 이 드라이버를 통해 **DSN(Data Source Name)**을 등록하여 데이터베이스 접속 정보를 저장한다.

DSN을 등록하는 방법은 다음과 같이 2가지가 있다.

- 사용자 DSN

현재 로그인한 Windows 계정에서만 사용할 수 있는 DSN이다.

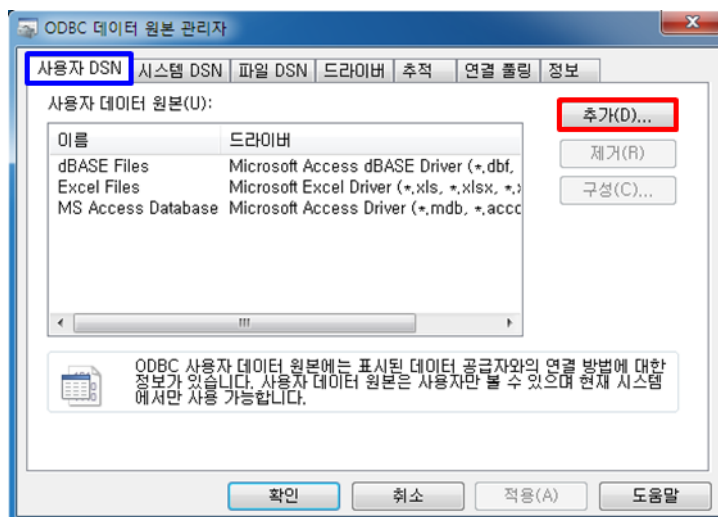
- 시스템 DSN

어떠한 Windows 계정으로도 사용할 수 있는 DSN이다.

등록 순서는 다음과 같다. 본 안내서에서는 **사용자 DSN**이라는 방법을 통해 DSN을 등록한다.

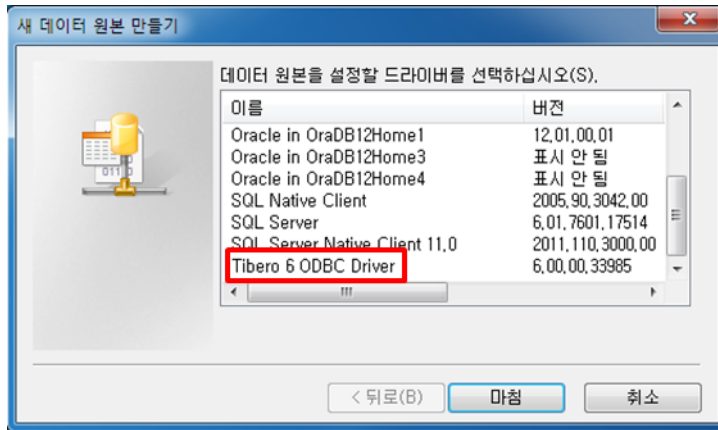
1. ODBC 데이터 원본 관리자를 실행한다.
2. **[사용자 DSN]** 탭으로 이동한 후 **[추가]** 버튼을 클릭한다.

[그림 5.2] 사용자 DSN - 추가



3. 설치된 Tibero 6 ODBC Driver를 선택한 후 **[마침]** 버튼을 클릭한다.

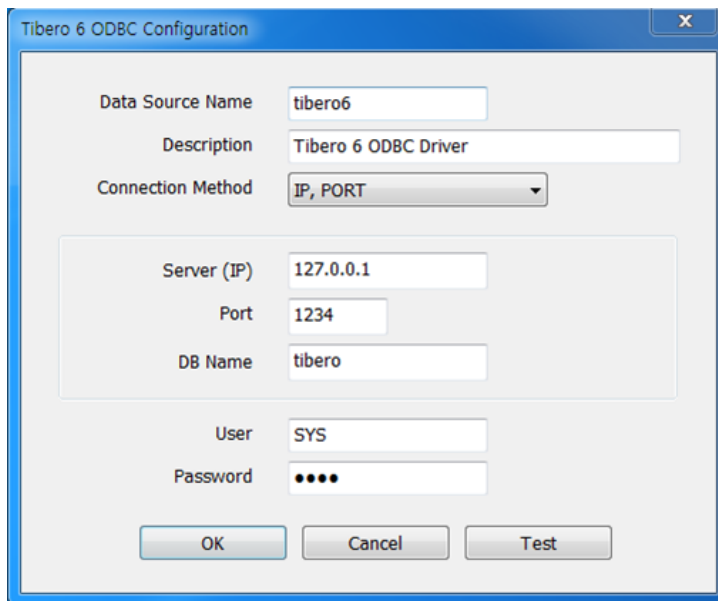
[그림 5.3] 사용자 DSN - Tibero 6 ODBC Driver 선택



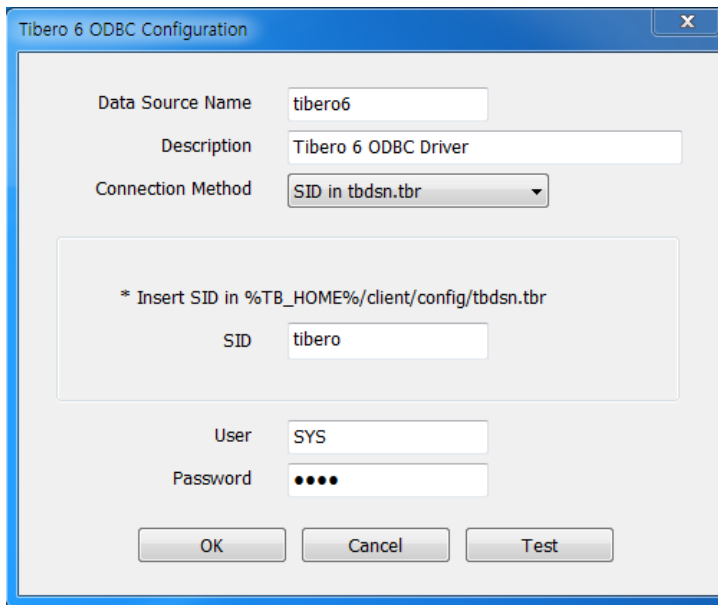
4. **TIBERO 6 ODBC Configuration** 대화상자가 나타난다. 이 대화상자에 데이터베이스 접속 정보를 입력한다.

TIBERO 6 ODBC Driver에서는 IP와 PORT로 접속하는 방식과 tbdns.tbr 파일에 설정된 SID로 접속하는 방식 두 가지를 제공한다(IP/PORT 방식으로 접속하면 TB_HOME 환경을 구축하지 않고도 간편하게 드라이버를 사용할 수 있지만 tbdns.tbr 파일에 TB_NLS_LANG 설정값을 넣는 등의 기능을 사용할 수 없다는 단점이 있다).

[그림 5.4] 사용자 DSN - 데이터베이스 접속 정보 입력(IP, PORT 방식)

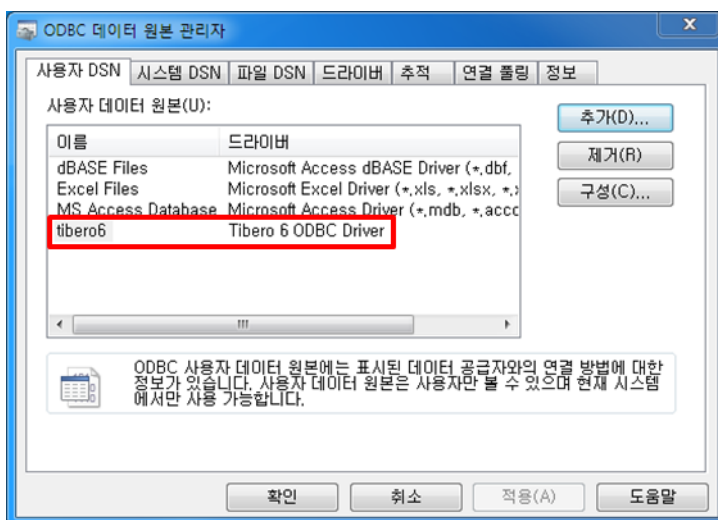


[그림 5.5] 사용자 DSN - 데이터베이스 접속 정보 입력(SID 방식)



5. 데이터베이스 접속 정보가 모두 입력되면 **[OK]** 버튼을 클릭한다. 등록이 완료되면 다음 그림과 같이 새로 등록된 DSN이 표시된다.

[그림 5.6] 사용자 DSN - 등록 완료



DSN 등록이 완료되면 **'tibero6'** 를 SQLConnect 또는 SQLDriverConnect 함수에 사용할 수 있으며, 이 함수를 통해 데이터베이스에 접속할 수 있다.

5.2.2. UNIX 계열에서의 설치

Windows 계열에서는 운영체제를 설치하면 자동으로 ODBC 드라이버 관리자가 설치되지만, UNIX 계열의 환경에서는 별도로 설치해야 한다. ODBC 드라이버 관리자는 unixODBC, iODBC 등 여러 종류가 있다.

참고

본 안내서에서는 iODBC 드라이버 관리자를 설치하는 방법을 설명한다.

iODBC는 다음의 웹사이트에서 다운로드할 수 있다.

```
http://iodbc.org/
```

운영체제에 직접 소스 코드를 다운로드한 후 Makefile을 실행해야 하므로, libiodbc-3.52.6.tar.gz 형태의 파일을 다운로드한다.

기본 환경 설정

iODBC를 설치하기 위해서는 다음과 같은 기본 환경을 설정해야 한다.

기본 환경을 설정하는 순서는 다음과 같다.

1. \$TB_HOME/client/lib(또는 \$TB_HOME/client/lib32) 디렉터리에 libtbodbc.so 파일이 존재하는지 확인한다. 확인하는 방법은 다음과 같다.

```
$ file libtbodbc.so
libtbodbc.so: ELF 32-bit LSB shared object, Intel 80386,
version 1 (SYSV), not stripped
```

2. LD_LIBRARY_PATH 환경변수에 libtbodbc.so 파일이 존재하는 경로를 추가한다. 단, 특정 머신에 따라 LD_LIBRARY_PATH가 아닌 다른 환경변수를 사용할 수도 있다. 예를 들어 AIX 5.3 머신에서는 LIBPATH라는 환경변수를 사용한다.

iODBC 설치

iODBC는 Tiberio의 설치 버전이 지원하는 CPU에 따라 다음과 같이 설치 방법이 달라진다.

- 32-bit

iODBC를 설치할 때 별도의 옵션을 주지 않으면, Tiberio의 설치 버전에 관계없이 32-bit로 iODBC가 설치된다. 예를 들어 다음과 같이 prefix 옵션을 주지 않는 경우에는 /usr/local 디렉터리에 iODBC가 설치된다.

```
export CC=cc
./configure [--prefix=/usr/local] --sysconfdir=/etc --disable-gui
```

```
make
make install
```

- 64-bit

iODBC는 기본적으로 32-bit로 컴파일된다. 따라서 64-bit 머신에서 동작 중인 Tiberο를 사용할 경우 반드시 컴파일 옵션을 별도로 주어 64-bit iODBC를 설치해야 한다.

다음은 머신별로 iODBC를 설치하는 방법이다.

- HP-UX

```
export CFLAGS=+DD64
export CC=cc
./configure [--prefix=/usr/local] --sysconfdir=/etc --disable-gui
make
make install
```

- AIX

```
export CFLAGS=-q64
export CC=cc
./configure [--prefix=/usr/local] --sysconfdir=/etc --disable-gui
make
make install
```

iODBC 설치 확인

iODBC를 설치한 후에는 **file** 명령을 사용하여 컴파일이 제대로 수행되었는지 확인한다.

다음은 설치된 iODBC가 컴파일을 제대로 수행하였는지 확인하는 예이다.

```
$ file libtbodbc.so
libtbodbc.so: ELF 32-bit LSB shared object, Intel 80386,
version 1 (SYSV), not stripped

$ file libiodbcinst.so.2.1.18
libiodbcinst.so.2.1.18: ELF 32-bit LSB shared object, Intel 80386,
version 1 (SYSV), not stripped
```

Tiberο를 위한 iODBC 환경설정

설치가 완료된 iODBC에 환경설정을 적용하는 순서는 다음과 같다.

1. ODBC 드라이버를 설정한다.

아래의 예를 참고하여 /etc 디렉터리에 odbc.ini 파일을 생성한다.

<odbc.ini>

```
[ODBC Data Source]
Tibero6      = Tibero 6 ODBC Driver

[ODBC]
Trace        = 1
TraceFile    = /tmp/odbc.trace

[Tibero6]
Driver       = /home/tibero6/client/lib/libtbodbc.so
Description  = Tibero6 ODBC Datasource
SID          = tibero ; tbdsn.tbr 파일에 설정한 DSN 정보
User        = tibero
Password     = tmax
```

odbc.ini 파일은 시스템 전체에 대한 설정을 의미한다. 사용자별로 별도의 ODBC 드라이버를 설정하고자 할 때에는 \$HOME/.odbc.ini 파일을 위 예와 같이 작성하면 된다.

2. ODBC 데이터소스를 설정한다.

아래의 예를 참고하여 /etc 디렉터리에 odbcinst.ini 파일을 생성한다.

<odbcinst.ini>

```
[Tibero 6 ODBC Driver]
Description = ODBC Driver for Tibero 6
Driver      = /home/tibero6/client/lib/libtbodbc.so
UsageCount  = 1
```

3. 1, 2번 과정을 통해 생성된 ODBC 드라이버와 ODBC 데이터소스를 iODBC 관리자에 등록한다.

등록하는 방법은 다음과 같다.

```
iodbc-config --odbc.ini --odbcinst.ini
```

데이터베이스 접속 확인

iODBC에 환경설정을 완료한 후 데이터베이스에 접속이 되는지 확인하려면 다음과 같은 명령을 실행해야 한다.

```
iodbctest "DSN={dsn};UID={user};PWD={pwd}"
```

항목	설명
dsn	DSN의 이름
user	접속할 사용자 계정의 이름
pwd	접속할 사용자 계정의 패스워드

다음은 **iodbctest** 명령을 사용하여 데이터베이스 접속을 확인하는 예이다.

```
$ iodbctest "DSN=Tibero6;UID=tibero;PWD=tmax"
iODBC Demonstration program
This program shows an interactive SQL processor
Driver Manager: 03.52.0607.1008
Driver: 06.00.0000 (libtbodbc.so)

SQL>select * from dual;

DUMMY
-----
X

result set 1 returned 1 rows.

SQL>
```

참고

iodbctest 명령을 실행할 때 데이터베이스 접속 정보가 /etc/odbc.ini 파일과 \$HOME/.odbc.ini 파일 양쪽 모두에 존재하는 경우에는 \$HOME의 경로를 먼저 탐색한다는 점에 유의해야 한다. 만약 ODBCINI 환경변수가 설정되어 있다면, ODBCINI 환경변수에 입력된 odbc.ini 파일을 가장 먼저 탐색한다.

5.3. tbdsn.tbr과 ODBC 데이터 원본 관리자

tbCLI에서는 다음과 같이 두 가지 방법으로 데이터베이스 접속 정보를 가져올 수 있다.

- tbdsn.tbr

```
tibero=(
    (INSTANCE=(HOST=12.34.56.78)
      (PORT=8629)
      (DB_NAME=tibero))
)
```

- ODBC 데이터 원본 관리자의 DSN

Windows 계열의 tbCLI 또는 UNIX 계열(LINUX 포함)에서 설치한 ODBC에서 데이터베이스 접속 정보를 가져올 때에는 우선 ODBC 데이터 원본 관리자를 검색한다.

만약 ODBC 데이터 원본 관리자에 해당 정보가 없다면 tbdsn.tbr 파일을 검색하여 데이터베이스 접속 정보를 찾는다. tbCLI를 사용하는 tbSQL 유틸리티에서도 이와 동일한 과정을 거쳐 데이터베이스 접속 정보를 가져온다.

색인

B

BINARY_DOUBLE, 10
BINARY_FLOAT, 10
BLOB, 11

C

Call Level Interface, 1
CHAR, 10
CLOB, 11
connection handle, 2

D

DATE, 11
descriptor handle, 2
DSN, 141, 147

E

environment handle, 2

F

FLOAT, 10

I

INTEGER, 10
INTERVAL DAY TO SECOND, 11
INTERVAL YEAR TO MONTH, 11
iODBC, 144

L

LONG, 10
LONG RAW, 10

N

NCHAR, 10
null-terminating string, 5

NUMBER, 10
NVARCHAR, 10

O

ODBC, 1, 139
Open Database Connectivity, 1, 139

R

RAW, 10
ROWID, 11

S

SQL 문장 실행 방법
 준비된 실행, 6
 직접 실행, 6
SQL_ERROR, 119
SQL_INVALID_HANDLE, 119
SQL_NEED_DATA, 119
SQL_NO_DATA, 119
SQL_NTS, 5
SQL_STILL_EXECUTING, 119
SQL_SUCCESS, 119
SQL_SUCCESS_WITH_INFO, 119
SQLAllocConnect (Deprecated), 13
SQLAllocEnv (Deprecated), 13
SQLAllocEnv2 (Deprecated), 102
SQLAllocHandle, 13
SQLAllocHandle2, 103
SQLAllocStmt (Deprecated), 15
SQLBindCol, 15
SQLBindParameter, 17
SQLCancel, 18
SQLCloseCursor, 19
SQLColAttribute, 20
SQLColAttributes (Deprecated), 22
SQLColumnPrivileges, 22
SQLColumns, 23
SQLConnect, 26
SQLCopyDesc, 28
SQLDescribeCol, 29
SQLDescribeParam, 31
SQLDisconnect, 32

SQLDriverConnect, 33
 SQLEndTran, 35
 SQLError (Deprecated), 37
 SQLExecDirect, 37
 SQLExecDirect 함수, 3
 SQLExecute, 40
 SQLExtendedFetch (Deprecated), 42
 SQLExtProcRaiseError, 42
 SQLExtProcRaiseErrorWithMsg, 43
 SQLFetch, 44
 SQLFetchScroll, 45
 SQLForeignKeys, 47
 SQLFreeConnect (Deprecated), 49
 SQLFreeEnv (Deprecated), 49
 SQLFreeHandle, 50
 SQLFreeStmt, 51
 SQLGetConnectAttr, 52
 SQLGetConnectOption (Deprecated), 53
 SQLGetCursorName, 53
 SQLGetData, 54
 SQLGetDescField, 56
 SQLGetDescRec, 58
 SQLGetDiagField, 4, 59
 SQLGetDiagRec, 4, 60
 SQLGetEnvAttr, 62
 SQLGetExtProcAllocMemory, 63
 SQLGetExtProcConnect, 64
 SQLGetFunctions, 65
 SQLGetInfo, 66
 SQLGetStmtAttr, 67
 SQLGetStmtOption (Deprecated), 68
 SQLGetTypeInfo, 68
 SQLGetWarningMsg, 105
 SQLGetWarningMsgLength, 105
 SQLLobClose, 106
 SQLLobFreeLoc, 107
 SQLLobGetData (Deprecated), 108
 SQLLobGetData2, 110
 SQLLobGetLength, 112
 SQLLobOpen, 113
 SQLLobPutData, 114
 SQLLobTruncate, 116
 SQLMoreResults, 70

SQLNativeSql, 71
 SQLNumParams, 72
 SQLNumResultCols, 73
 SQLParamData, 74
 SQLParamOptions (Deprecated), 76
 SQLPrepare, 76
 SQLPrimaryKeys, 77
 SQLProcedureColumns, 79
 SQLProcedures, 80
 SQLPutData, 82
 SQLRETURN, 3, 119
 SQLRowCount, 84
 SQLSetConnectAttr, 85
 SQLSetConnectOption (Deprecated), 86
 SQLSetCursorName, 86
 SQLSetDescField, 87
 SQLSetDescRec, 89
 SQLSetEnvAttr, 91
 SQLSetParam (Deprecated), 92
 SQLSetPos, 92
 SQLSetScrollOptions (Deprecated), 94
 SQLSetStmtAttr, 94
 SQLSetStmtOption (Deprecated), 95
 SQLSpecialColumns, 95
 SQLSTATE, 3
 00000, 02000, 138
 01000 ~ 01007, 120
 01S00 ~ 01S09, 121
 07001 ~ 07009, 122
 07S01, 122
 08001 ~ 08007, 123
 08S01, 123
 21S01 ~ 21S02, 124
 22001 ~ 22026, 124
 23000, 24000, 25000, 126
 25S01 ~ 25S03, 126
 28000, 34000, 127
 3C000 ~ 3F000, 127
 40001 ~ 40003, 42000, 128
 42S01 ~ 42S22, 44000, 128
 HY000 ~ HY024, 129
 HY090 ~ HY111, 132
 HYC00, 135

HYT00 ~ HYT01, 135

IM001 ~ IM015, 136

SQLStatistics, 97

SQLTablePrivileges, 99

SQLTables, 101

SQLTransact (Deprecated), 102

statement handle, 2

T

tbCLI, 1

tbCLI 특징, 1

tbCLI 프로그램 구조, 4

SQL 문장 실행 및 에러 처리 부분, 6

시작 설정 부분, 5

종료 설정 부분, 7

tbCLI의 데이터 타입, 11

tbdsn.tbr, 147

Tibero의 데이터 타입, 9

TIME, 11

TIMESTAMP, 11

TIMESTAMP WITH LOCAL TIME ZONE, 11

TIMESTAMP WITH TIME ZONE, 11

V

VARCHAR, 10

ㄱ

간격형, 10

ㄴ

날짜형, 9

내재형, 10

ㄷ

대용량 객체형, 10

데이터소스, 5

ㅁ

멀티 스레드, 103

문자형, 9

ㅂ

반환 코드 목록, 119

ㅅ

사용자 DSN, 141

상태 레코드, 3

숫자형, 9

스케일, 10

시스템 DSN, 141

ㅈ

정밀도, 10

진단 레코드, 3

ㅎ

핸들, 2

문장 핸들, 2

서술자 핸들, 2

연결 핸들, 2

환경 핸들, 2

헤더 레코드, 3

