



테이블 만들기 및 다양한 데이터 유형 학습

Database Fundamentals

발표자 이름

날짜

© 2019, Amazon Web Services, Inc. 또는 계열사. All rights reserved.

테이블 만들기 및 다양한 데이터 유형 학습을 시작하겠습니다.

교육 내용

이 강의의 핵심

배울 내용은 다음과 같습니다.

- 기존 데이터베이스에서 새 테이블 만들기
- 테이블 생성 시 데이터 유형 구현하기

주요 용어:

- 데이터 조작 언어(DML)
- 데이터 정의 언어(DDL)
- 데이터 제어 언어(DCL)
- 사전 정의된 데이터 유형
- 구성된 데이터 유형
- 사용자 정의된 데이터 유형
- 숫자 데이터 유형
- 문자열 유형
- 기본 키(PK)
- 외래 키(FK)



2

aws re/start

이 모듈에서 학습할 내용은 다음과 같습니다.

- 기존 데이터베이스에서 새 테이블 만들기
- 테이블 생성 시 데이터 유형 구현하기

관계형 데이터베이스의 구조

관계형 데이터베이스

- 유용한 작업을 수행할 수 있도록 정형 데이터 모음

테이블

- 스프레드시트와 유사함
- 특정 개념에 대한 관련 정보 모음

필드 또는 속성

- 열이라고도 함
- 하나의 열에는 한 가지 유형의 정보(예: 정수, 텍스트, 통화 값)가 포함됨

레코드

- 행이라고도 함
- 하나의 행에는 테이블에 있는 한 엔터티에 대한 정보가 포함됨



데이터베이스



Customers



Products



Orders

Id	Image	Title	location
1	chain_saw.png	Chainsaw	Aisle 5, Bin G5
2	circular_saw.png	Circular Saw	Aisle 5, Bin G2
3	hacksaw.png	Hacksaw	Aisle 7, Bin A2
4	hand_saw.png	Hand Saw	Aisle 7, Bin A3

관계형 데이터베이스는 사전 정의된 관계가 있는 데이터 항목의 모음입니다.

데이터 베이스의 항목은 열과 행으로 이루어진 테이블 집합으로 구성됩니다. 테이블은 데이터베이스에 표시할 해당 객체들에 관한 정보를 수록하는 데 사용됩니다. 테이블의 각 열은 특정 데이터를 수록하며 필드는 속성의 실제 값을 저장합니다. 테이블의 행은 한 객체 또는 엔터티와 관련된 값의 모음을 나타냅니다. 테이블의 각 행은 기본 키라고 부르는 고유 식별자로 표시할 수 있고 여러 테이블에 있는 행은 외래 키를 사용하여 연결할 수 있습니다.

다음 슬라이드에서는 관계형 데이터베이스, 데이터베이스 구축에 사용되는 언어 및 데이터베이스의 일부에 대해 설명합니다.

정형 쿼리 언어(SQL)

정형 쿼리 언어 또는 **SQL**(see-quell 또는 ess-cue-ell로 발음)은 데이터 쿼리 및 조작, 데이터베이스의 구조 정의에 사용되는 언어입니다. SQL은 관계형 데이터베이스를 위한 표준 프로그래밍 언어입니다.

SQL, 강력하면서도 간단한 언어

SQL의 기능은 무엇입니까?

- 기존 데이터베이스에서 새 테이블 만들기
- 테이블 생성 시 데이터 유형 구현하기

하위 언어 그룹

- 데이터 조작 언어(DML)
- 데이터 정의 언어(DDL)
- 데이터 제어 언어(DCL)

SQL의 기능은 무엇입니까?

SQL에서는 **SQL 스테이트먼트**를 사용하여 데이터베이스에서 필요한 많은 작업을 수행합니다. 서로 다른 SQL 스테이트먼트는 SQL의 하위 언어 아래에 집합적으로 그룹으로 구성됩니다. SQL 그룹의 하위 언어는 다음과 같습니다.

- 데이터 조작 언어(DML)
- 데이터 정의 언어(DDL)
- 데이터 제어 언어(DCL)

SQL 스테이트먼트 그룹

스테이트먼트 그룹	스테이트먼트 그룹 설명	스테이트먼트
데이터 조작 언어 (DML)	<ul style="list-style-type: none"> 테이블의 데이터 보고 변경하고 조작함 테이블에 데이터를 선택, 업데이트 및 삽입하고 테이블에서 데이터를 삭제하는 명령이 포함됨 일반적으로 클라이언트 애플리케이션을 작성하는 데이터 분석가, 보고서 작성자 및 프로그래머가 사용함 	SELECT UPDATE INSERT DELETE
데이터 정의 언어 (DDL)	<ul style="list-style-type: none"> 데이터베이스와 포함된 개체를 생성 및 정의함 테이블 생성 및 삭제 명령이 포함됨 일반적으로 데이터베이스 관리자 및 프로그래머가 사용함 	CREATE TABLE DROP TABLE
데이터 제어 언어 (DCL)	<ul style="list-style-type: none"> 데이터베이스의 데이터에 대한 액세스 권한을 제어함 데이터베이스 권한을 부여하거나 취소하는 명령이 포함됨 일반적으로 데이터베이스 관리자 및 프로그래머가 사용함 	REVOKE GRANT

SQL 키워드

데이터 조작 언어(DML)

SELECT – 테이블에서 데이터 검색

INSERT – 테이블에 행 추가

UPDATE – 테이블에 있는 행 수정

DELETE – 테이블에서 행 삭제

데이터 정의 언어(DDL)

CREATE – 데이터베이스 또는 테이블 생성

ALTER TABLE – 테이블의 열 추가, 삭제 또는 수정, 또는 제약 조건 추가 또는 삭제

DROP – 테이블 또는 제약 조건과 같은 데이터베이스 개체 삭제

데이터 제어 언어(DCL)

REVOKE – 데이터베이스 사용자의 권한 취소

GRANT – 데이터베이스 사용자의 권한 부여

첫 번째 스테이트먼트 그룹은 데이터 조작 언어(DML)입니다. DML을 사용하면 테이블의 데이터를 보거나 추가하거나 변경하거나 삭제할 수 있습니다.

두 번째 스테이트먼트 그룹은 데이터 정의 언어(DDL)입니다. DDL을 사용하면 데이터베이스(해당 스키마)의 개체를 정의하고 유지 관리할 수 있습니다. 스키마에는 데이터베이스의 테이블, 열 및 데이터 유형이 포함됩니다.

세 번째 스테이트먼트 그룹은 데이터 제어 언어(DCL)입니다. DCL 스테이트먼트는 데이터베이스의 데이터에 대한 액세스를 제어합니다.



SQL 요소

이 섹션에서는 SQL 스테이트먼트의 구성 요소인 SQL 언어의 여러 빌딩 블록을 살펴봅니다.

데이터 유형

데이터 유형은 데이터베이스의 데이터를 조작하는 데 사용할 수 있습니다. 데이터 유형은 스토리지 형식, 데이터의 값 및 작업을 결정합니다.

- **사전 정의된** 데이터 유형 또는 **기본 제공 데이터 유형**
- **생성된** 유형은 SQL 데이터 유형의 생성자를 사용하여 지정됨
- **사용자 정의된** 데이터 유형은 표준으로 정의할 수 있음

데이터 유형은 데이터베이스를 조작하는 데 사용할 수 있습니다. 데이터 유형은 스토리지 형식, 데이터 유형의 값 및 작업을 결정합니다. 각 데이터 유형은 표현 가능한 값의 모음입니다. 모든 값은 하나 이상의 데이터 유형에 속합니다. SQL에는 사전 정의된 유형, 구성된 유형 및 사용자 정의된 유형의 세 가지 데이터 유형이 있습니다.

독점적인 데이터 형식

표준 SQL은 광범위한 데이터 유형을 제공합니다.

데이터베이스 관리 시스템(DBMS)은 추가 독점 데이터 유형을 제공할 수도 있습니다.

이러한 데이터 유형은 다른 시스템에서 지원되지 않을 수 있습니다.

서로 다른 시스템 간의 호환성을 높이려면 가능한 한 표준 데이터 유형을 사용하십시오.

식별자에 적용되는 대소문자 규칙

식별자는 언어 키워드나 스테이트먼트와 달리 사용자가 만드는 항목을 나타냅니다. 식별자는 소문자로 작성합니다.

키워드, 명령 및 스테이트먼트는 대문자로 작성할 수 있습니다.

AWS re/Start에서 작업할 때는 모든 식별자가 소문자여야 합니다.

그러나 다른 DBMS는 요구 사항에 따라 이 규칙을 다르게 처리합니다.

IBM 및 Oracle

작성한 코드를 처리할 때 IBM 및 Oracle DBMS는 자동으로 식별자를 대문자로 변환합니다(즉, 사용자가 사용한 대소문자를 무시함). 식별자에 사용한 대소문자를 유지하려면 큰따옴표(" ")로 묶어야 합니다.

Microsoft SQL Server

Microsoft SQL Server는 대소문자를 구분하거나 구분하지 않도록 구성할 수 있지만 기본적으로 대소문자를 구분합니다. 대소문자 구분은 데이터의 정렬 규칙, 대소문자, 악센트 구분 속성을 결정하는 SQL Server의 데이터 정렬 속성과 연결됩니다.

MySQL Server

MySQL Server는 Microsoft Windows를 제외하고 기본적으로 대소문자를 구분합니다.



숫자 데이터 유형

이 섹션에서는 SQL 스테이트먼트의 구성 요소인 SQL 언어의 여러 빌딩 블록을 알아봅니다.

숫자 유형

숫자 데이터 유형은 숫자 값을 나타냅니다.

데이터 유형	설명
INTEGER	정수를 나타냅니다. 최소값 및 최대값은 DBMS에 따라 다릅니다.
SMALLINT	DBMS에 따라 범위가 더 작은 값을 보유할 수 있다는 점을 제외하고는 INTEGER 유형과 동일합니다.
BIGINT	DBMS에 따라 범위가 더 큰 값을 보유할 수 있다는 점을 제외하고는 INTEGER 유형과 동일합니다.
DECIMAL(p, s)	정확한 수치, 정밀도는 p , 스케일은 s 입니다. 소수점을 가질 수 있는 숫자인 10진수입니다. 크기 인수는 정밀도와 스케일의 두 부분으로 구성됩니다. 스케일은 정밀도를 초과할 수 없습니다.
NUMERIC(p, s)	정확한 수치, 정밀도는 p , 스케일은 s 입니다. 최대 정밀도는 DBMS에 따라 다릅니다.
FLOAT(p)	대략적 수치, 가수 정밀도 p . 정밀도는 1 이상이며 최대 정밀도는 DBMS에 따라 다릅니다.
REAL	DBMS가 정밀도를 정의한다는 점을 제외하면 FLOAT 유형과 동일합니다.

15

숫자 데이터 유형은 숫자 값을 나타냅니다. **정확한 숫자 데이터 유형**이라고 하는 경우도 있습니다. 정확한 숫자 데이터 형식은 가능한 모든 값을 안정적으로 포함할 수 있는 가장 작은 데이터 형식을 사용합니다.

정수 유형은 정확한 숫자를 나타내며 일반적으로 테이블의 기본 키로 사용합니다.

INTEGER를 사용할 때 저장할 수 있는 정수 값의 범위를 식별합니다. 다음을 준수해야 합니다.

- 저장할 모든 값을 수용할 수 있을 만큼 충분히 큰 범위를 선택합니다.
- 스토리지 및 폐기물 스토리지를 과도하게 할당하지 않습니다.

자세한 내용은 [SQL 데이터 유형](#)을 참조하십시오.

통화 값



4639.00
8031.00



통화 값을 FLOAT 또는 REAL 값 대신 **DECIMAL** 값으로 저장합니다.

통화 값은 MySQL Server와 같은 일부 시스템에서 특별 데이터 유형을 사용하여 지원됩니다. 이러한 데이터 유형은 DECIMAL보다 데이터를 더 효율적으로 저장하며 통화 기호를 값과 연결하는 기능을 포함합니다.

예상치 못한 결과를 얻지 않도록 통화 값에 대한 계산을 수행할 때 항상 주의하십시오.

FLOATING POINT, DECIMAL, REAL

FLOAT 및 REAL 데이터 유형:

DECIMAL 값보다 더
효율적으로 저장됨

일반적으로 DECIMAL
값보다 빠르게 처리할
수 있음

효율적이고 빠르지만 항
상 정확하지는 않음

빠르고 정확해야 하는 특
정 과학 계산과 같은 작업
에 적합함

숫자가 정확해야 하는 회
계 등의 작업에는 적합하
지 않음

날짜 및 시간 데이터

- **날짜(DATE):** 날짜를 나타냅니다. 예: yyyy-mm-dd
- **시간(TIME):** 표준 시간대 없이 하루 중 시간을 나타냅니다. 예: hh:mm:ss
- **이벤트(TIMESTAMP):** 트랜잭션이 발생한 날짜 및 시간과 같은 특정 시점을 나타냅니다.
 - 예: 2020년 6월 24일 오전 8시 30분(동부 표준시)
- **기간(INTERVAL):** 특정 날짜 및 시간의 컨텍스트를 벗어난 기간을 나타냅니다.
 - 예: 60분, 초 없음(60:00)



문자열 유형

이 섹션에서는 SQL 언어의 여러 요소에 대해 알아봅니다.

문자열

이 데이터 유형은 **문자열** 데이터 유형 설명자로 설명됩니다.

데이터 유형	설명
CHARACTER (CHAR)	<ul style="list-style-type: none">문자열, 고정 길이구현자 정의된 형식의 텍스트 문자열이 유형의 값은 작은따옴표(')로 묶어야 함
CHARACTER VARYING (VARCHAR)	<ul style="list-style-type: none">가변 길이 문자열, 최대 길이 고정
CHARACTER LARGE OBJECT (CLOB)	<ul style="list-style-type: none">문자 대용량 객체(CLOB)는 DBMS의 문자 데이터 모음CLOB는 일반적으로 테이블 자체에서 참조되는 별도의 위치에 저장됨

문자열 데이터 유형은 **문자열** 데이터 유형 설명자로 설명됩니다.

문자열에 대한 몇 가지 핵심 사항

- 사용하는 시스템에서 텍스트 스토리지가 할당되는 방식을 이해하고 있는지 확인하십시오.
- 우편 번호, 제품 코드 또는 전화 번호와 같은 일관된 길이 데이터에 **길이가 고정된 문자 데이터 유형**을 사용합니다.
- 데이터 길이의 가변성이 클 때 가변 길이 데이터 유형을 사용합니다.
 - 가변 길이 열이 필요한 것보다 길지 않은지 확인합니다.
 - 예: **사용자 ID**는 20자를 초과할 수 없습니다.

키워드 및 식별자

이 섹션에서는 SQL 언어의 여러 요소에 대해 알아봅니다.

SQL 키워드

키워드는 SQL이 예약한 단어임

키워드는 SQL이 예약하고 사전 정의된 데이터 유형이 포함된 단어입니다.

- SELECT, GRANT, DELETE, CREATE

식별자는 데이터베이스의 개체 또는 항목에 대한 목적에 맞는 이름입니다. 이러한 항목은 다음과 같습니다.

- 테이블, 열, 별칭, 인덱스 및 보기

키워드는 SQL이 예약하고 사전 정의된 데이터 유형이 포함된 단어입니다. 특정 컨텍스트 범위를 벗어나는 키워드를 사용하면 오류가 발생합니다. 키워드의 몇 가지 예가 나와 있습니다.

식별자는 테이블, 열, 별칭, 인덱스, 보기, 기타 개체 및 데이터베이스 자체와 같은 데이터베이스 개체에 지정되는 목적에 맞는 이름입니다.

MySQL에서 지원되는 데이터 유형

정확한 수치

- SMALLINT
- INTEGER
- BIGINT
- DECIMAL
- NUMERIC
- REAL

대략적 수치

- DOUBLE PRECISION
- FLOAT

바이너리

- BINARY
- BINARY VARYING
- BINARY LARGE OBJECT

부울

- BOOLEAN

문자열

- CHARACTER
- CHARACTER VARYING
- CHARACTER LARGE OBJECT
- NATIONAL CHARACTER
- NATIONAL CHARACTER VARYING
- NATIONAL CHARACTER LARGE OBJECT

날짜 및 시간

- DATE
- DATETIME WITHOUT TIMEZONE
- TIMESTAMP WITHOUT TIMEZONE
- TIME WITH TIMEZONE
- TIMESTAMP WITH TIMEZONE

간격

- INTERVAL DAY
- INTERVAL YEAR

모음

- ARRAY
- MULTiset

기타 유형

- ROW
- XML

제약 조건

이 섹션에서는 제약 조건, 기본 키(PK) 및 외래 키(FK)에 대해 알아봅니다.

제약 조건은 테이블에 들어갈 수 있는 데이터 유형을 제한하는 규칙입니다. 예를 들어 테이블의 열에 기본 키 제약 조건을 적용할 수 있습니다. 이 제약 조건을 적용하면 열이 테이블의 각 행을 고유하게 식별함을 나타냅니다. 결과적으로 데이터베이스 시스템은 이 열의 값이 고유하고 테이블의 모든 행에서 NULL이 아님을 확인합니다.

제약 조건

제약 조건은 테이블에 들어갈 수 있는 데이터 유형에 제한을 적용합니다.

NOT NULL:

열에 NULL 값이 없도록 합니다.

UNIQUE:

열 또는 열 집합에 해당 열에
고유한 값이 있어야 함

DEFAULT:

레코드에 대해 값이 정의되지 않은
경우 DEFAULT는 DBMS가
레코드를 테이블에 삽입할 때 값을
제공함

열에 저장할 수 있는 값
을 허용하거나 제한하는
규칙을 정의하는 데 사
용됨

테이블 생성 시 선언함

테이블에 입력할 수 있
는 데이터의 종류를 제
한하는 데 사용됨

데이터베이스의 신뢰성을
강화하기 위함

입력 및 편집 제약 조건

열에 다음 제약 조건을 설정할 수 있습니다.

- Required
 - 사용자는 해당 열에 유효한 값이 입력될 때까지 행을 저장할 수 없습니다.
 - NULL 값은 사용할 수 없습니다.
- No duplicates
 - 해당 열의 값이 이미 다른 행에 있는 경우 사용자는 행을 저장할 수 없습니다.
 - 이 제약 조건은 열에 대한 인덱스 설정과 함께 설정되는 경우가 많습니다.

기본 키(PK) 및 외래 키(FK)

이 섹션에서는 기본 키(PK) 및 외래 키(FK)에 대해 알아봅니다.

기본 키(PK) 및 외래 키(FK)

고객	
PK	customerid int NOT NULL
	familyName varchar(50) NOT NULL
	givenName varchar(50) NOT NULL
	streetAddress varchar(100) NOT NULL
	city varchar(50) NOT NULL
	zipCode varchar(20) NOT NULL
	INDEX(customerid, customerName)

외래 키

다른 테이블의 기본 키 값이 있는 테이블의 특수 열입니다. 두 테이블 간에 관계를 만듭니다.

기본 키

일반적으로 숫자인 각 행에 대해 고유한 값이 있는 테이블의 특수 열입니다.

주문	
PK	orderid int NOT NULL
FK1	customerid int NOT NULL
	orderDate datetime NOT NULL

입력 및 편집 제약 조건

기본 키 열에는 자동으로 다음 두 가지 제약 조건이 적용됩니다.

- 사용하는 관계형 DBMS에 따라 특정 데이터 유형에 대한 **입력 마스크** 또는 **그림**을 정의할 수 있습니다.
 - 예: 날짜는 다양한 방법으로 입력할 수 있습니다. 일반적으로 **1553224273**(UNIX Epoch time)은 2019년 3월 22일 금요일 오전 3:11:13 또는 2019-03-22 03:11:13(YYYY-MM-DD HH:MM:SS)에 해당합니다.
 - **입력 마스크**는 사용자를 안내하고 일관된 값이 열에 입력되도록 합니다.
 - 입력 마스크는 형식을 적용하는 데 도움이 될 수 있습니다. 예를 들어, 미국 사회 보장 번호의 형식은 **nnn-nn-nnnn**입니다. 입력 마스크는 사회 보장 번호에 포함되는 대시가 올바른 위치에 오도록 해줍니다.
 - 입력 마스크를 사용하여 고객 번호 열에서 특정 패턴의 문자와 숫자가 뒤따르도록 할 수 있습니다.
- 열에 값을 입력한 후 값의 편집을 금지할 수도 있습니다.



테이블

테이블 이름 지정하기

목적에 맞는 명명 규칙

목적에 맞게 이름(식별자)을 신중하게 선택합니다.
특정 요소에 따라 명명 규칙을 결정되어야 합니다.

- 예를 들어 데이터베이스, 테이블 및 열의 경우 다음을 관찰합니다.
 - DBMS가 부과하는 규칙 및 제한 사항
 - 조직에서 채택한 명명 규칙
 - 명확성

IF NOT EXISTS 절을 추가하는 것은 이름 지정 규칙이 이미 사용 중이 아닌지 확인하기 위한 선택 사항이지만 도움이 됩니다.

CREATE

테이블은 SQL에서 논리적으로 구성된 데이터 스토리지 단위입니다.

다음 방법으로 테이블을 생성합니다.

- **CREATE** 스테이트먼트

```
CREATE TABLE employees (
```

다음 방법으로 테이블이 있는지 확인합니다.

- **IF NOT EXISTS**가 포함된 **CREATE** 스테이트먼트

```
CREATE TABLE IF NOT EXISTS employees (
```

스테이트먼트는 **CREATE**로 시작하며 이는 **테이블을 생성한다는** 의미입니다. 이제 테이블이라고 선언합니다. 선언한 후에는 **IF NOT EXISTS** 명령을 사용하여 테이블 이름이 다른 이름과 중복되지 않도록 합니다.

CREATE TABLE

샘플 코드

```
CREATE TABLE employees (  
  employee_id INTEGER NOT NULL PRIMARY KEY,  
  given_name varchar(20) DEFAULT NULL,  
  family_name varchar(25) NOT NULL,  
  dept_id INTEGER NOT NULL  
);
```

이 SQL 스테이트먼트는 이름이 **employees**인 테이블을 생성합니다. 테이블에는 **employee_id**, **given_name**, **family_name**, **dept_id**의 4개 열이 있습니다. **employee_id** 열은 삽입 시 중복된 값이 포함되어 있지 않으며 기본 키로 사용됩니다. 또한 **employee_id**는 NOT NULL입니다(필드에 값이 포함되어야 함). **given_name** 및 **family_name** 필드는 varchar(문자 데이터의 가변 문자 필드 집합)입니다. **employee_id**는 INTEGER(숫자)입니다.

테이블 및 식별자

필드	Type	Null	기본값
employee_id	INTEGER	NO	NOT NULL
last_name	varchar (20)	NO	NOT NULL
first_name	varchar (25)	YES	DEFAULT
dept_id	INTEGER	NO	NOT NULL

```
1 CREATE TABLE employees (  
2   employee_id INTEGER NOT NULL PRIMARY KEY,  
3   first_name varchar(20) DEFAULT NULL,  
4   last_name varchar(25) NOT NULL,  
5   dept_id INTEGER NOT NULL  
6 );
```

- 각 열에는 사용할 수 있는 특정 데이터 유형과 최대 길이가 있습니다.
- 테이블에 여러 열을 추가할 때는 열을 공백과 쉼표로 구분합니다.

```
column_name DATA_TYPE (length) [NOT NULL] [DEFAULT value]
[auto_increment]
```

참조 무결성

참조 무결성: NULL이 아닌 모든 외래 키 값이 기존 기본 키 값과 일치하는 데이터베이스 품질



Employees

employee_id	family_name	given_name	dept_id
EN1-10	Wang	Xiulan	SL
EN1-12	Ramirez	Diego	AC
EN1-15	Wolf	Nikki	MK
EN1-16	Stiles	John	RI
EN1-19	Major	Mary	IS
EN1-20	Santos	Paulo	SL
EN1-22	Sarkar	Saanvi	MK
EN1-23	Martinez	Sofia	RI
EN1-25	Manu	Kwesi	IS
EN1-26	Roe	Jane	MK
EN1-27	Mansa	Akua	SL
EN1-28	Liu	Jie	AC
EN1-30	Jayashankar	Nikhil	RI
EN1-33	Silva	Ana Carolina	IS

Departments

dept_id	dept_name
AC	계정 관리
IS	정보 시스템
MK	마케팅
RI	접수 및 재고
SL	영업

학습 내용 확인 질문



데이터베이스 내부에 테이블을 만드는 데 사용되는 SQL 키워드는 무엇입니까?



데이터베이스에서 기본 키의 용도는 무엇입니까?

정답:

1. 키워드 CREATE는 데이터베이스에 테이블을 만드는 데 사용됩니다.
2. 기본 키는 테이블의 각 행에 대한 고유 식별자를 만드는 데 사용됩니다.

핵심 사항



© 2020, Amazon Web Services, Inc. 또는 계열사. All rights reserved.

37

- **식별자**는 언어 키워드나 스테이트먼트와 달리 사용자가 만드는 항목을 나타냅니다.
- **키워드**는 SQL이 예약하고 사전 정의된 데이터 유형이 포함된 단어입니다. 범위를 벗어난 키워드를 사용하면 오류가 발생합니다.
- **참조 무결성**은 NULL이 아닌 모든 외래 키 값이 기존 기본 키 값과 일치하는 데이터베이스 품질입니다.

aws re/start

이 강의에서 다룬 핵심 사항은 다음과 같습니다.

- 식별자는 언어 키워드나 스테이트먼트와 달리 사용자가 만드는 항목을 나타냅니다.
- 키워드는 SQL이 예약하고 사전 정의된 데이터 유형이 포함된 단어입니다. 범위를 벗어난 키워드를 사용하면 오류가 발생합니다.
- 참조 무결성은 NULL이 아닌 모든 외래 키 값이 기존 기본 키 값과 일치하는 데이터베이스 품질입니다.