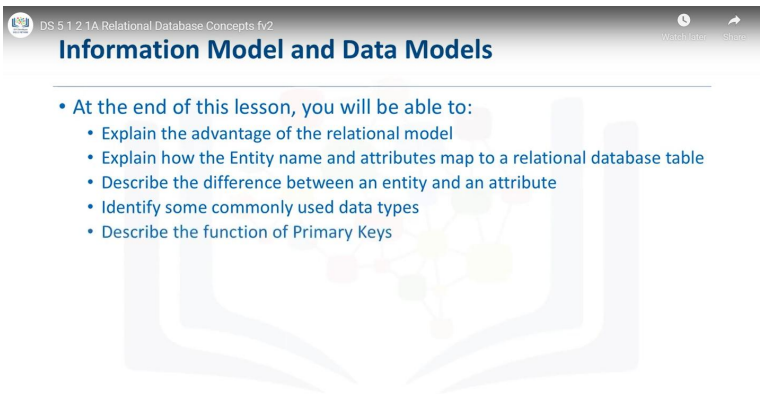
Module 2 - Introduction to Relational Databases and Tables

In this module, you will explore the fundamental concepts behind databases, tables, and the relationships between them. You will then create an instance of a database, discover SQL statements that allow you to create and manipulate tables, and then practice them on your own live database.

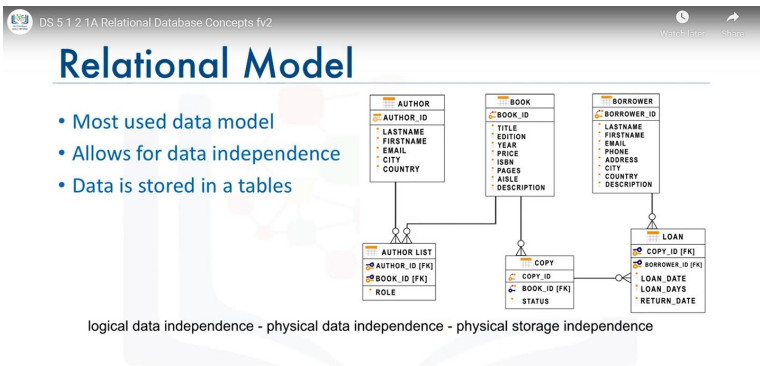
Learning Objectives

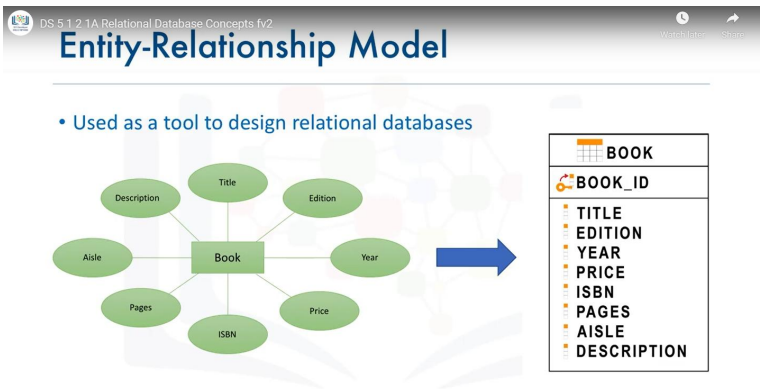
- Describe basic relational database concepts including tables, primary keys, and foreign keys.
- Create a database instance on the Cloud
- Distinguish between Data Definition Language and Data Manipulation Language.
- Explain the syntax of the CREATE TABLE statement.
- Explain the syntax of the ALTER, DROP, and TRUNCATE statements.
- Compose and execute CREATE TABLE, ALTER, DROP, and TRUNCATE statements hands-on on a live database.

2.1 Relational Database Concepts



In this video, we will learn about different types of models, how we use models to map data to tables, and define relationships between tables. At the end of this lesson, you will be able to explain the advantage of the relational'model, explain how the entity name and attributes map to a relational database table, describe the difference between an entity and an attribute, identify some commonly used data types, and describe the function of primary keys.

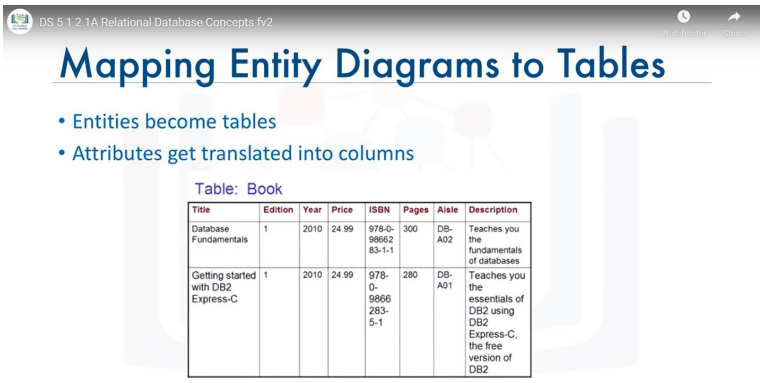


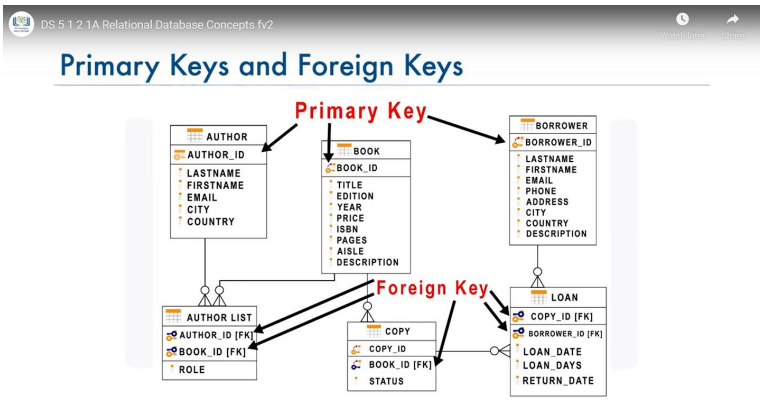


The **relational model** is the most used data model for databases because this model allows for data independence. Data is stored in a simple data structure. Tables: this provides logical data independence, physical data independence, and physical storage independence. An entity relationship data model, or ER data model, is an alternative to a relational data model.
Using a simplified library database as an example, this figure shows an entity relationship diagram or ERD that represents entities called tables and their relationships. In the library example, we have books. A book can be written by one or many authors. The library can have one or many copies of a book. Each copy can be borrowed by only one borrower at a time.

An **entity relationship model** proposes thinking of a database as a collection of entities rather than being used as a model on its own. The **ER model** is used as a tool to design relational databases. In the ER model, entities are objects that exist independently of any other entities in the database. The building blocks of an ER diagram are entities and attributes. An entity can be a noun: person, place, or thing. In an ER diagram, an entity is drawn as a rectangle. Entities have attributes which are the data elements that characterize the entity. Attributes tell us more about the entity. In an ER diagram, attributes are drawn as ovals.
Using a simplified library as an example, the book is an example of an entity. Attributes are certain properties or characteristics of an entity and tell us more about the entity. The entity book has attributes such as book title, the edition of the book, the year the book was written, etc. Attributes are connected to exactly 1 entity. The entity book becomes a table in the database and the attributes become the columns in a table.

总结:

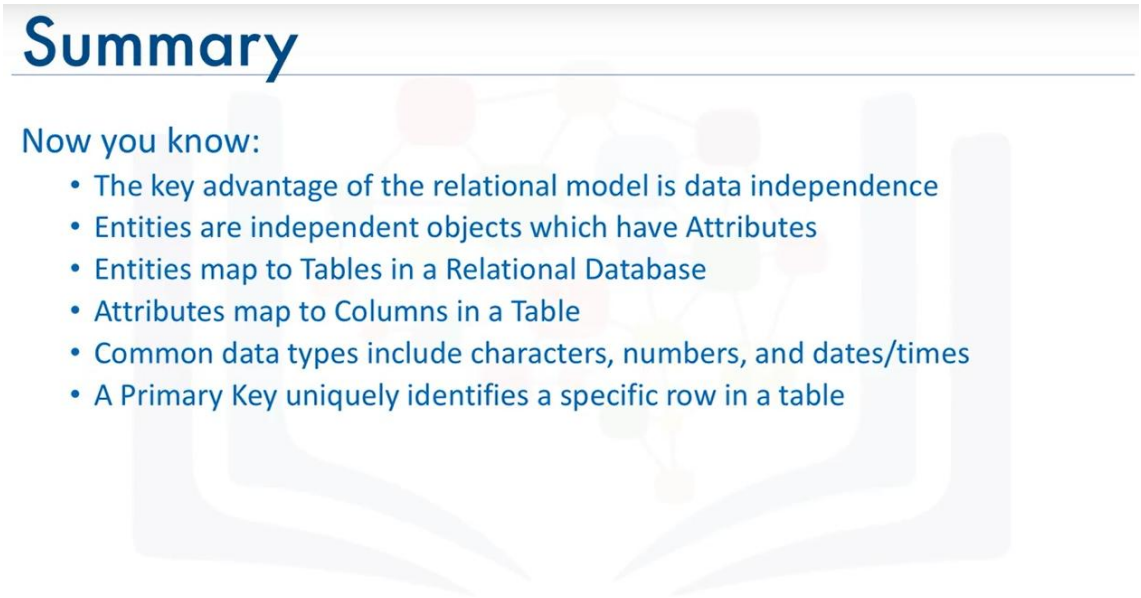




A table is a combination of rows and columns. While mapping, the entity becomes the table. Having said that, the table has not yet taken the form of rows and columns. The attributes get translated into columns in a table providing the actual table form of rows and columns. Later, we add some data values to each of the columns, which completes the table form. Each attribute stores data values of different formats, characters, numbers dates, currency, and many more besides.
In the book table example, the title is made up of characters. As book titles vary in length, we can set the variable character data type for the title column: Varchar. For character columns that do not vary in length, we use character or Char. The Edition and year columns would be numeric. The ISBN column would be Char because it contains dashes as well as numbers and so on.

Using the book entity mapping as an example, we can create the tables for the remainder of our simplified library example using entity names, like author, author list, borrower, loan, and copy. The entity attributes will be the columns of the tables. Each table is assigned a primary key.

- **The primary key** of a relational table uniquely identifies each tuple or row in a table, preventing duplication of data and providing a way of defining relationships between tables.
- Tables can also contain **foreign keys** which are primary keys defined in other tables, creating a link between the tables.

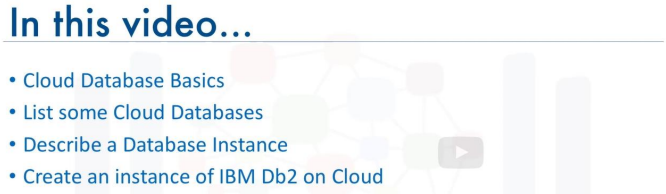


Now you know that:

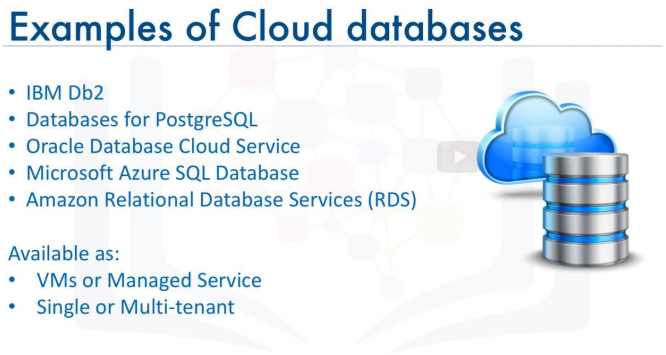
- the key advantage of the relational model is logical and physical data independence and storage independence.
- Entities are independent objects which can have multiple characteristics called attributes.
- When mapping to a relational database, entities are represented as tables and attributes map to columns.
- Common data types include characters such as Char and Varchar, numbers such as integer and decimal, and timestamps including date and time.
- A primary key uniquely identifies a specific row in a table and prevents duplication of data.

总结:

2.2 How to Create a Database Instance on Cloud



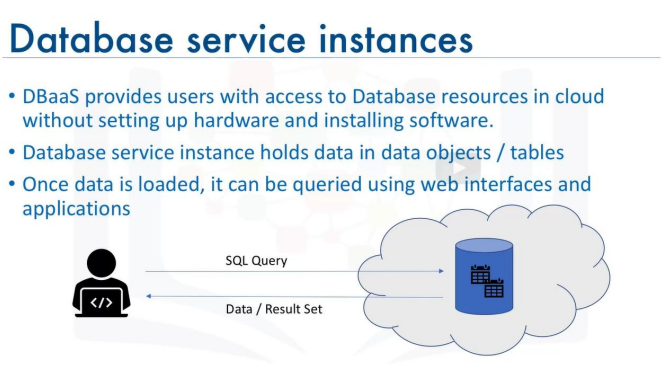
This video will cover the key concepts around databases in the Cloud. In order to learn SQL, you first need to have a database available to practice your SQL queries. An easy way to do so is to create an instance of a database in the Cloud and use it to execute your SQL queries.
After completing this lesson, you will be able to understand basic concepts related to Cloud databases, list a few Cloud databases, describe database service instances, as well as demonstrate how to create a service instance on an IBM Db2 on Cloud.



A Cloud database is a database service built and accessed through a Cloud platform. It serves many of the same functions as traditional databases with the added flexibility of Cloud computing. Some advantages of using Cloud databases include:

- Ease of use, users can access Cloud databases from virtually anywhere using a vendor's API or web interface, or your own applications whether on Cloud or Remote.
- Scalability. Cloud databases can expand and shrink their storage and compute capacities during runtime to accommodate changing needs and usage demands, so organizations only pay for what they actually use.
- Disaster recovery. In the event of a natural disaster or equipment failure or power outage, data is kept secure through backups on Remote Servers on Cloud in geographically distributed regions.

A few examples of relational databases on Cloud include, IBM Db2 on Cloud, databases for PostgreSQL on IBM Cloud, Oracle Database Cloud Service, Microsoft Azure SQL Database, and Amazon Relational Database Services. These Cloud databases can run in the Cloud either as a Virtual Machine, which you can manage, or delivered as a Managed Service depending on the vendor. The database services can either be single or multi-tenant depending on the service plan.

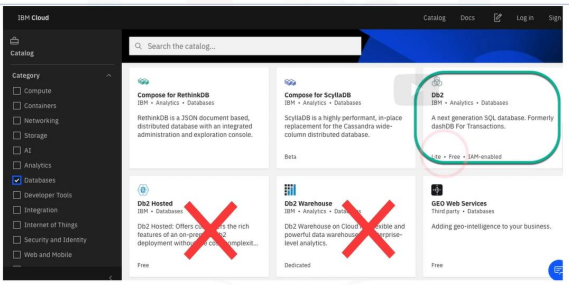


To run a database in Cloud, you must first provision an instance of the database service on the Cloud platform of your choice. An instance of a **Database-as-a-Service** or **DBaaS** provides users with access to database resources in Cloud without the need for setting up of the underlying hardware, installing the database software, and administering the database. The database service instance will hold your data in related tables. Once your data is loaded into the database instance, you can connect to the database instance using a web interface or APIs in your applications. Once connected, your application can send SQL queries across to the database instance. The database instance then resolves the SQL statements into operations against the data and objects in the database. Any data retrieved is returned to the application as a result set.

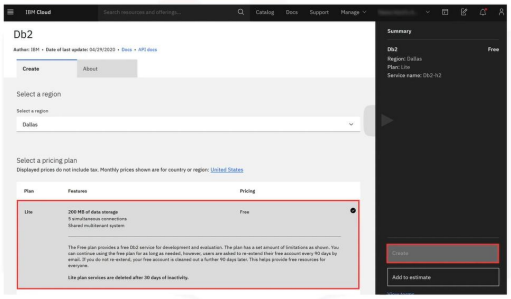
Now let's see how a database instance is created for Db2 on Cloud. IBM Db2 on Cloud is a SQL database provisioned for you in the Cloud. You can use Db2 on Cloud just as you would use any database software, but without the overhead and expensive hardware setup or software installation and maintenance.

总结:

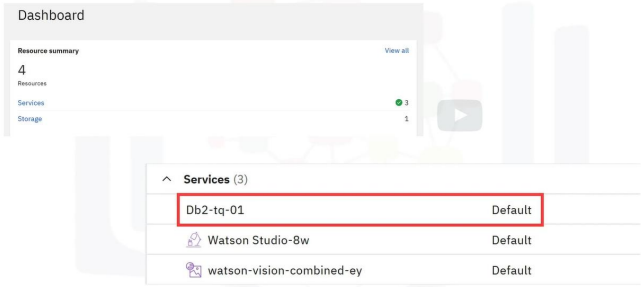
Deploy an instance of Db2 on Cloud Service



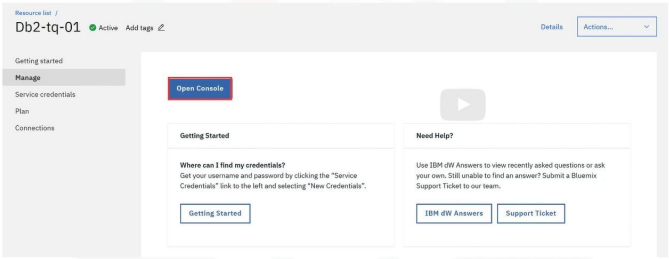
Create a new service



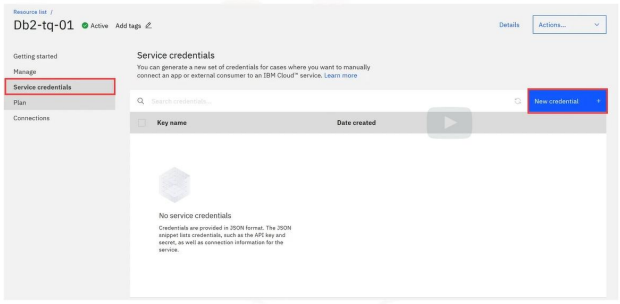
View the newly created service



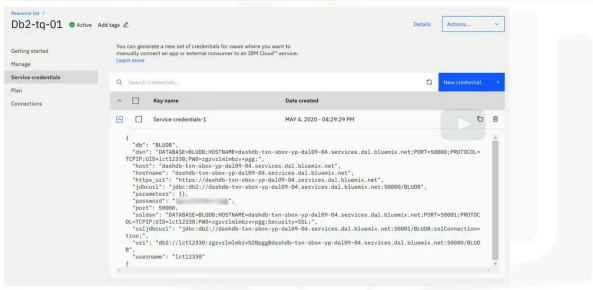
Manage the database instance



Create new service credentials



Service credentials



Service credentials

The credentials include the necessary details to establish a connection to the database, and includes the following;

- a database name
- port number
- a host name, which is the name of the server on the Cloud on which your database instance resides,
- a username, which is the user ID you'll use to connect along with the password. Note that your username is also the schema name in which your tables will be created by default.

Now let's see how we can set up a service instance of Bb2.

1. Navigate to IBM Cloud catalog and select the Db2 service. Note there are several variations of the Db2 service, including Db2 Hosted and Db2 Warehouse. For our purposes, we will choose the Db2 service which comes with a free lite plan. Select the lite plan.
2. If need to, change the defaults. You can type a service instance name, choose the region to deploy to, as well as an org and space for the service, then click "Create".
3. You can view the IBM Db2 service that you created by selecting services from your IBM Cloud dashboard.
4. From this dashboard, you can manage your database instance. For example, you can click on the "Open Console" button to launch the Web Console for your database instance. The Web Console allows you to create tables, load data, explore data in your tables, and issue SQL queries.
5. In order to access your database instance from your applications, you will need the service credentials. For the first time around, you'll need to create a set of new credentials. You can also choose to create multiple sets of credentials for different applications and users.
6. Once a set of service credentials is created, you can view it as adjacent snippet.

Now that you know how to create a database instance on Cloud, the next step is to actually go and create one.

总结:

2.3 Types of SQL Statements (DDL vs. DML)

Welcome to Types of SQL Statements.

At the end of the video, you will be able to distinguish between data definition language statements and data manipulation language statements.

SQL Statements are used for interacting with Entities (that is, tables), Attributes (that is, columns) and their tuples (or rows with data values) in relational databases. SQL statements fall into two different categories:

- **Data Definition Language statements**
- **Data Manipulation Language statements**

Data Definition Language (or DDL) statements are used to define, change, or drop database objects such as tables.
Common DDL statement types include CREATE, ALTER, TRUNCATE, and DROP.

- CREATE: which is used for creating tables and defining its columns;
- ALTER: is used for altering tables including adding and dropping columns and modifying their datatypes;
- TRUNCATE: is used for deleting data in a table but not the table itself;
- DROP: is used for deleting tables.

Data Manipulation Language (or DML) statements are used to read and modify data in tables. These are also sometimes referred to as CRUD operations, that is, Create, Read, Update and Delete rows in a table.
Common DML statement types include INSERT, SELECT, UPDATE, and DELETE.

- INSERT: is used for inserting a row or several rows of data into a table;
- SELECT: reads or selects row or rows from a table;
- UPDATE: edits row or rows in a table;
- And DELETE: removes a row or rows of data from a table.

Now you know that:

- DDL or Data Definition Language statements are used for defining or changing objects in a database such as tables.
- DML or Data Manipulation Language statements are used for manipulating or working with data in tables.

总结:

2.4 CREATE TABLE Statement

SKILLS NETWORK

CREATE TABLE Statement

IBM Developer

Objectives

At the end of this video, you will be able to:

- Create a Table in a relational database using Entity Name, Attributes and the CREATE TABLE statement

CREATE table

- **Syntax:**

```
CREATE TABLE table_name
   (
    column_name_1 datatype optional_parameters,
    column_name_2 datatype,
    ...
    column_name_n datatype
   )
```

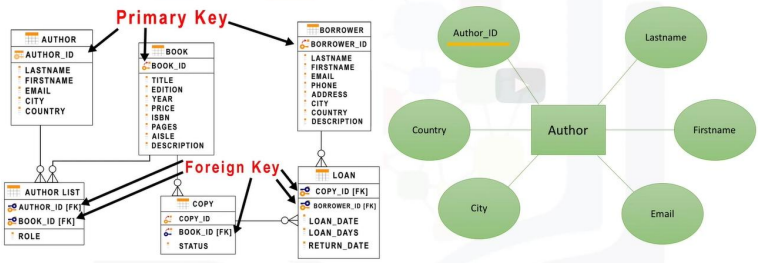
EXAMPLE

- Create a table for Canadian provinces

```
CREATE TABLE provinces(
    id char(2) PRIMARY KEY NOT NULL,
    name varchar(24)
   )
```

id *char(2)*	name *varchar(24)*
AB	ALBERTA
BC	BRITISH COLUMBIA
...	...

Create a table



Primary Key: Uniquely Identifies each Row in a Table

CREATE TABLE Statement

To create the Author table, use the following columns and datatypes:

AUTHOR(Author_ID:char, Lastname:varchar, Firstname:varchar, Email:varchar, City:varchar, Country:char)

```
CREATE TABLE author (
    author_id CHAR(2) PRIMARY KEY NOT NULL,
    lastname VARCHAR(15) NOT NULL,
    firstname VARCHAR(15) NOT NULL,
    email VARCHAR(40),
    city VARCHAR(15),
    country CHAR(2)
   )
```

with constaint NOT NULL, ensuring that they can not contain a null value.

Summary

Now you know that:

- CREATE used for creating entities (tables) in a relational database
- CREATE TABLE statement includes definition of attributes (columns):
  - Names of columns
  - Datatypes of columns
  - Constraints (e.g. Primary Key)

总结:

2.5 ALTER, DROP, and Truncate Tables

ALTER, DROP, and TRUNCATE Tables

IBM Developer

Objectives

After watching this video, you will be able to:

- Describe the ALTER TABLE, DROP TABLE, and TRUNCATE statements
- Explain the syntax
- Use the statement in queries

Hello and welcome to ALTER, DROP, and TRUNCATE tables. After watching this video, you will be able to:

- Describe the ALTER TABLE, DROP TABLE, and TRUNCATE statements.
- Explain the syntax.
- Use the statements in queries.

ALTER TABLE ... ADD COLUMN

- Add or remove columns
- Modify the data type of columns
- Add or remove keys
- Add or remove constraints

```
ALTER TABLE <table_name>
  ADD COLUMN <column_name_1> datatype
  . . .
  ADD COLUMN <column_name_n> datatype;
```

ALTER TABLE ... ADD COLUMN

```
ALTER TABLE author
  ADD COLUMN telephone_number BIGINT;
```

author_id	lastname	firstname	email	city	country	telephone_number
1001	Thomas	John	johnt@...	New York	USA	5551111
1002	James	Alice	alicej@...	Seattle	USA	5551112
1003	Wells	Steve	stevew:@...	Montreal	Canada	5552222
1004	Kumar	Santosh	kumars@...	London	UK	5553333

You use the **ALTER TABLE statement** to add or remove columns from a table, to modify the data type of columns, to add or remove keys, and to add or remove constraints. The syntax of the ALTER TABLE statement is shown here. You start with ALTER TABLE followed by the name of the table that you want to alter. Differently to the CREATE TABLE statement though, you do not use parentheses to enclose the parameters for the ALTER TABLE statement. Each row in the ALTER TABLE statement specifies one change that you want to make to the table.

For example, to add a telephone number column to the AUTHOR table in the Library database to store the author's telephone number, use the following statement: **ALTER TABLE author ADD COLUMN telephone_number BIGINT;** In this example, the data type for the column is BIGINT which can hold a number up to 19 digits long.

ALTER TABLE ... ALTER COLUMN

```
ALTER TABLE <table_name>
  ALTER COLUMN <column_name> SET DATA TYPE
<datatype>;
```

ALTER TABLE ... ALTER COLUMN

```
ALTER TABLE author
  ALTER COLUMN telephone_number SET DATA TYPE
CHAR(20);
```

author_id	lastname	firstname	email	city	country	telephone_number
1001	Thomas	John	johnt@...	New York	USA	555-1111
1002	James	Alice	alicej@...	Seattle	USA	555-1112
1003	Wells	Steve	stevew@...	Montreal	Canada	555-2222
1004	Kumar	Santosh	kumars@...	London	UK	555-3333

You also use the ALTER TABLE statement to modify the data type of a column. To do this, use the ALTER COLUMN clause specifying the new data type for the column.

For example, using a numeric data type for telephone number means that you cannot include parentheses, plus signs, or dashes as part of the number. You can change the column to use the CHAR data type to overcome this. This code shows how to alter the author table: **ALTER TABLE author ALTER COLUMN telephone_number SET DATA TYPE CHAR(20);**

Altering the data type of a column containing existing data can cause problems though if the existing data is not compatible with the new data type. For example, changing a column from the CHAR data type to a numeric data type will not work if the column already contains non-numeric data. If you try to do this, you will see an error message in the notification log and the statement will not run.

总结:

ALTER TABLE ... DROP COLUMN

```
ALTER TABLE author
  DROP COLUMN telephone_number;
```

author_id	lastname	firstname	email	city	country	telephone_number
1001	Thomas	John	johnt@...	New York	USA	555-1111
1002	James	Alice	alicej@...	Seattle	USA	555-1112
1003	Wells	Steve	stevew:@...	Montreal	Canada	555-2222
1004	Kumar	Santosh	kumars@...	London	UK	555-3333

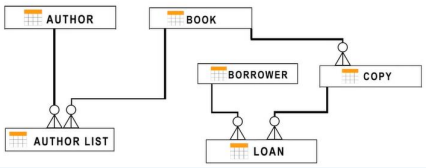
If your spec changes and you no longer need this extra column, you can again use the ALTER TABLE statement, this time with the DROP COLUMN clause, to remove the column as shown:

ALTER TABLE author DROP COLUMN telephone_number;

DROP TABLE

```
DROP TABLE <table_name>;
```

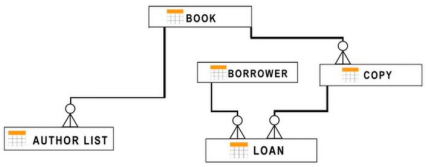
```
DROP TABLE author;
```



DROP TABLE

```
DROP TABLE <table_name>;
```

```
DROP TABLE author;
```



Similar to using DROP COLUMN to delete a column from a table, you use the **DROP TABLE statement** to delete a table from a database. If you delete a table that contains data, by default the data will be deleted alongside the table. The syntax for the DROP TABLE statement is: **DROP TABLE table_name ;**
So, you use this statement: **DROP TABLE author;** to remove the table from the database.

TRUNCATE TABLE

```
TRUNCATE TABLE author
  IMMEDIATE;
```

author_id	lastname	firstname	email	city	country
1001	Thomas	John	johnt@...	New York	USA
1002	James	Alice	alicej@...	Seattle	USA
1003	Wells	Steve	stevew:@...	Montreal	Canada
1004	Kumar	Santosh	kumars@...	London	UK

Sometimes you might want to just delete the data in a table rather than deleting the table itself.
While you can use the DELETE statement without a WHERE clause to do this, it is generally quicker and more efficient to truncate the table instead. You use the TRUNCATE TABLE statement to delete all of the rows in a table. The syntax of the statement is: **TRUNCATE TABLE table_name IMMEDIATE;**
The IMMEDIATE specifies to process the statement immediately and that it cannot be undone.

So, to truncate the author table, you use this statement:
TRUNCATE TABLE author IMMEDIATE;

Summary

In this video, you learned that:

- The ALTER TABLE statement changes the structure of an existing table, for example to add, modify, or drop columns
- The DROP TABLE statement deletes an existing table
- The TRUNCATE TABLE statement deletes all rows of data in a table

In this video, you learned that:
- The ALTER TABLE statement changes the structure of an existing table, for example, to add, modify, or drop columns.
- The DROP TABLE statement deletes an existing table.
- The TRUNCATE TABLE statement deletes all rows of data in a table.

总结:

How does the syntax of a CREATE statement look?

```
CREATE TABLE table_name (
    column1 datatype,
    column2 datatype,
    column3 datatype,
   ....
);
```

How does the syntax of an ALTER statement look?

```
ALTER TABLE table_name
ADD COLUMN column_name data_type column_constraint;

ALTER TABLE table_name
DROP COLUMN column_name;

ALTER TABLE table_name
ALTER COLUMN column_name SET DATA TYPE data_type;

ALTER TABLE table_name
RENAME COLUMN current_column_name TO new_column_name;
```

How does the syntax of a TRUNCATE statement look?

```
TRUNCATE TABLE table_name;
```

How does the syntax of a DROP statement look?

```
DROP TABLE table_name;
```

总结:

Summary & Highlights

Congratulations! You have completed this lesson. At this point in the course, you know:

- A database is a repository of data that provides functionality for adding, modifying, and querying the data.
- SQL is a language used to query or retrieve data from a relational database.
- The Relational Model is the most used data model for databases because it allows for data independence.
- The primary key of a relational table uniquely identifies each tuple or row, preventing duplication of data and providing a way of defining relationships between tables.
- SQL statements fall into two different categories: Data Definition Language (DDL) statements and Data Manipulation Language (DML) statements.

总结: