



# IT-234 – database concepts

UNIT 4 – USING SQL COMMANDS TO MODIFY THE DATABASE SCHEMA

# overview



You put a lot of thought into your design for the database.



You refined the logical model again and again until it was just right; then you implemented it.



That is when you found it — a mistake!

# overview

Not to worry, we can change the schema of the database after it is created.

In this unit, you will explore some of the SQL commands that you can use to alter the schema of a database.

# overview

After completing this unit, you should be able to:

- Apply SQL Data Definition Language (DDL) commands to modify an existing database schema.

# Database Life cycle

The database life cycle consists of four phases: requirements analysis, design, implementation, and maintenance.

The requirement analysis specifies relevant data to the users entirely and accurately.

The analysis/design phases consist of conceptual data modeling, logical design, and physical design.

# Database Life cycle

In the implementation phase, a database can be created using SQL or features existing in the DBMS or applicable tools.



The database must be maintained to ensure that evolving information requirements are met.

### Database maintenance

- Ensure that evolving information requirements are met
- Add, delete, or changes characteristics of the structure of a database in order to:
  - meet changing business conditions
  - correct errors
  - improve performance.
- Fix errors and recover database when it is contaminated

### Database implementation

- Create and test the database
- Complete database documentation and training materials
- Install database and convert data from prior systems

### Enterprise modeling

- Analyze current data processing
  - Analyze the general business functions and their database needs

### Conceptual data modeling

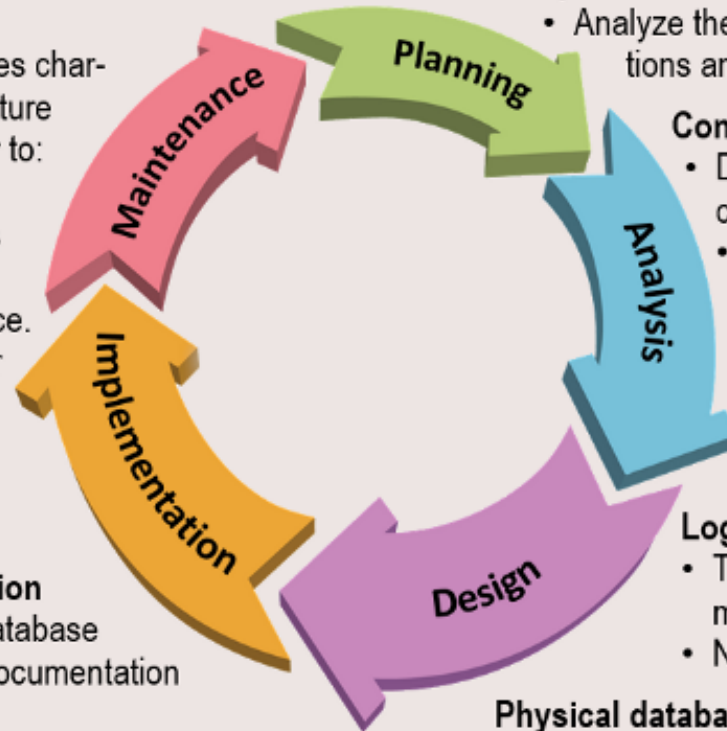
- Develop preliminary conceptual data model.
- Compare preliminary conceptual data model with enterprise data model
- Develop detailed conceptual data model

### Logical database design

- Transform conceptual data model into relations
- Normalization

### Physical database design

- Specify the organization of physical records, the choice of file organizations, and the use of indexes



# Database life cycle

# Database maintenance

- ▶ When the database comes into operation, monitoring is carried out to see if performance requirements are being met
  - Whether user expectations increase with demands for better performance and/or more database features
  - If not, modifications must be made to improve performance and/or add database features



# Database maintenance

- ▶ Database administrators perform routine database maintenance
  - Preventive maintenance (backup)
  - Corrective maintenance (restoration)
  - Adaptive maintenance (improves performance, adds entities and attributes, and so on)
  - Assignment of access permissions and their maintenance for new and old users

# Database maintenance

- ▶ Database administrators perform routine database maintenance
  - Database access statistics to improve efficiency and usability of system audits and to monitor system performance
  - Periodic security audits based on system-generated statistics
  - Summarize usage of the monthly, quarterly, or yearly system for internal billing or budgeting purposes

## Database maintenance



Modifications may be required as needs change.



The life cycle continues with monitoring, redesign, and improvement.

# Structured query language (SQL)

Structured Query Language (SQL) is the language in which we can perform operations on relational database systems

- We can use this language to create a database.

SQL uses certain commands like Create, Drop, Insert, etc. to carry out the required tasks.

# Structured query language (SQL)

- ▶ SQL commands are mainly categorized into five categories:
  - Data Definition Language (DDL)
  - Data Manipulation Language (DML)
  - Data Query Language (DQL)
  - Data Control Language (DCL)
  - Transaction Control Language (TCL)



DDL consists of the SQL commands that can be used to define the database schema.



It simply deals with descriptions of the database schema and is used to create and modify the structure of database objects in the database.

# Data Definition Language (DDL)

# Data Definition Language (DDL)

- ▶ DDL commands:
  - CREATE – used to create the database or its objects (like tables, indexes, functions, views, stored procedures, and triggers).
  - DROP – used to delete objects from the database.
  - ALTER - used to alter the structure of the database.
  - TRUNCATE – used to remove all records from a table
  - COMMENT – used to add comments to the data dictionary.
  - RENAME – used to rename an object existing in the database.

# Data manipulation Language (dml)



DML allows you to modify the database instance by inserting, modifying, and deleting its data.



DML is responsible for performing all types of data modification in a database.



# Data manipulation Language (DML)

- ▶ DML commands:
  - INSERT – used to insert data into a table.
  - UPDATE – used to update existing data within a table.
  - DELETE – used to delete records from a database table.

# Data Query Language (dql)

Data Query Language (DQL) is used to fetch the data from the database.

DQL uses only one command:

- SELECT - used to retrieve data from the database.

# Data control Language (dCL)

- ▶ DCL includes commands such as GRANT and REVOKE which mainly deal with the rights, permissions and other controls of the database system.
  - GRANT - gives users database object and action privileges.
  - REVOKE - withdraws user privileges.

# transaction control Language (tCL)

TCL commands deal  
with transactions  
within a database.



## TCL commands:

COMMIT –  
commits a  
Transaction.

ROLLBACK –  
rollbacks a  
transaction in  
case of any  
error occurs.

SAVEPOINT –  
sets a  
savepoint  
within a  
transaction.

SET  
TRANSACTION  
– specify  
characteristics  
for the  
transaction.

# DDL – CREATE TABLE



Tables are used to store data in the database.



Tables are uniquely named within a database and schema.



Each table contains one or more columns.



Each column has an associated data type that defines the kind of data it can store e.g., numbers, strings, or temporal data.

TO CREATE A NEW TABLE, YOU USE THE CREATE TABLE STATEMENT AS FOLLOWS:

```
CREATE TABLE [database_name.][schema_name.]table_name (  
    pk_column data_type PRIMARY KEY,  
    column_1 data_type NOT NULL,  
    column_2 data_type,  
    ...,  
    table_constraints  
);
```

# DDL – CREATE TABLE

# DDL – ALTER TABLE

The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.

The ALTER TABLE statement is also used to add and drop various constraints on an existing table.

# DDL – ALTER TABLE

- ▶ The following ALTER TABLE ADD statement appends a new column to a table:

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

- ▶ In this statement:
  - First, specify the name of the table in which you want to add the new column.
  - Second, specify the name of the column, its data type, and constraint if applicable.



# DDL – ALTER TABLE

- ▶ If you want to add multiple columns to a table at once using a single ALTER TABLE statement, you use the following syntax:

```
ALTER TABLE table_name
ADD
    column_name_1 data_type_1 column_constraint_1,
    column_name_2 data_type_2 column_constraint_2,
    ...,
    column_name_n data_type_n column_constraint_n;
```

- ▶ In this syntax, you specify a comma-separated list of columns that you want to add to a table after the ADD clause.

# DDL – ALTER TABLE

To modify the data type of a column, you use the following statement:

```
ALTER TABLE table_name  
ALTER COLUMN column_name new_data_type(size);
```

The new data type must be compatible with the old one, otherwise, you will get a conversion error in case the column has data and it fails to convert.

# DDL – ALTER TABLE

Sometimes, you need to remove one or more unused or obsolete columns from a table.

To do this, you use the ALTER TABLE DROP COLUMN statement as follows:


```
ALTER TABLE table_name  
DROP COLUMN column_name;
```

# DDL – ALTER TABLE

With ALTER TABLE DROP COLUMN syntax:

First, specify the name of the table from which you want to delete the column.

Second, specify the name of the column that you want to delete.



SQL Server does not allow you to delete a column that has a PRIMARY KEY or a FOREIGN KEY constraint.

# DDL – ALTER TABLE

If you want to delete multiple columns at once, you use the following syntax:

```
ALTER TABLE table_name  
DROP COLUMN column_name_1, column_name_2,...;
```

In this syntax, you specify columns that you want to drop as a list of comma-separated columns in the DROP COLUMN clause.

Sometimes, you want to remove a table that is no longer in use.

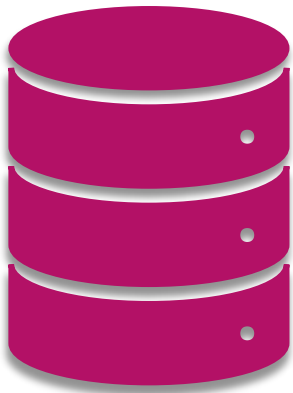
To do this, you use the following DROP TABLE statement:

```
DROP TABLE [IF EXISTS] [database_name.][schema_name.]table_name;
```

When SQL Server drops a table, it also deletes all data, triggers, constraints, permissions of that table.

DDL – DROP TABLE

# DDL – DROP TABLE



- ▶ With the DROP TABLE syntax:
  - Specify the name of the table to be removed.
  - Specify the name of the database in which the table was created and the name of the schema to which the table belongs.
  - The database name is optional → if you skip it, the DROP TABLE statement will drop the table in the currently connected database.

## DDL – DROP TABLE

- ▶ The IF EXISTS option is available from SQL Server 2016 (13.x).
  - It allows you to conditionally delete a table only if the table already exists.
  - If you attempt to delete a non-existing table without specifying the IF EXISTS option, SQL Server will issue an error.



## DDL – DROP TABLE

SQL Server does not explicitly drop the views and stored procedures that reference the dropped table.

To explicitly drop these dependent objects, you must use the DROP VIEW and DROP PROCEDURE statement.

# DDL – DROP TABLE

SQL Server does not allow you to delete a table that is referenced by a foreign constraint.

To delete this table, you must drop the referencing foreign key constraint or referencing table first.

# DDL – DROP database


To remove an existing database from a SQL Server instance, you use the DROP DATABASE statement.

The DROP DATABASE statement allows you to delete one or more databases with the following syntax:

```
DROP DATABASE [ IF EXISTS ]  
    database_name  
    [,database_name2,...];
```

# DDL – DROP database

With DROP DATABASE, you specify the name of the database that you want to drop after the DROP DATABASE keywords.



The IF EXISTS option is available from SQL Server 2016 (13.x).

It allows you to conditionally delete a database only if the database already exists.

If you attempt to delete a non-existing database without specifying the IF EXISTS option, SQL Server will issue an error.

# DDL – DROP database

- ▶ Before deleting a database, you must ensure the following important points:
  - ▶ First, the DROP DATABASE statement deletes the database and also the physical disk files used by the database.
    - Therefore, you should have a backup of the database in case you want to restore it in the future.
  - ▶ Second, you cannot drop the database that is currently being used.