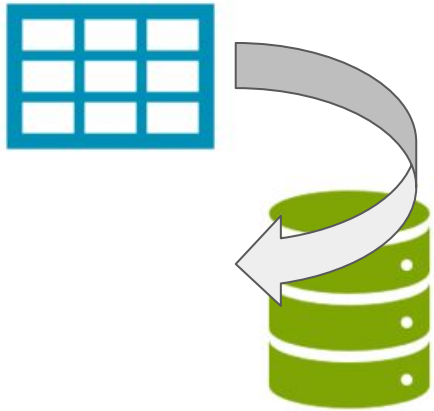
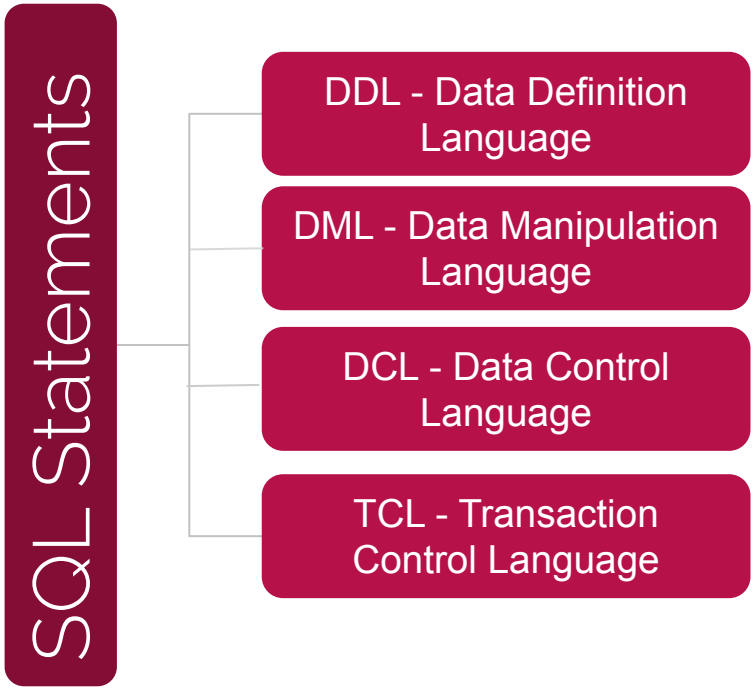
SQL
Session 5



DDL Commands

Table of Contents

Introduction



Data Definition Language

- DDL specifies the database schema.
- Some statements used in DDL are **CREATE**, **ALTER**, **DROP**.
- DDL statements are typically used to set up and configure a new database before we insert data.

Data Manipulation Language

- Data Manipulation Language (DML) enables users to access or manipulate data.
- **INSERT**, **UPDATE**, **DELETE**, **SELECT*** are the statements used in DML.

* In some sources, SELECT statement is grouped into a different category called DQL (Data Query Language).

Data Control Language

- Data Control Language (DCL) is used to grant or revoke access control.
- Its statements are **`REVOKE`** and **`GRANT`**.

Transaction Control Language

- Transaction Control Language (TCL) controls the transactions of DML and DDL commands.
- Some statements in TCL are **COMMIT, ROLLBACK, SAVEPOINT**.

2 Data Types

Data Types

The data type of a column defines what value the column can hold: integer, character, date and time, binary, and so on.

Data Types

String

Numeric

Date and Time

String Data Types

The string data types are:

- CHAR
- VARCHAR
- BINARY
- VARBİNARY
- BLOB
- TEXT

Date and Time Data Types

The date and time data types are:

- DATE
- DATETIME
- YEAR

Numeric Data Types

Integer Types (Exact Value)

- INTEGER or INT
- SMALLINT
- TINYINT
- MEDIUMINT
- BIGINT

Floating-Point Types (Approximate Value)

- FLOAT
- DECIMAL
- NUMERIC
- REAL

Data Types

Data types might have different names in different database. And even if the name is the same, the size and other details may be different! Always check the documentation!

3 CREATE TABLE

CREATE TABLE

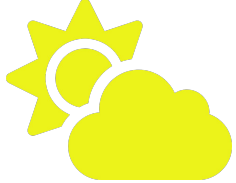
When creating a table, we use **CREATE TABLE** statement.

Syntax of a Basic Create Table Statement

```
CREATE TABLE table_name
        (column_name1 data_type,
         column_name2 data_type);
```

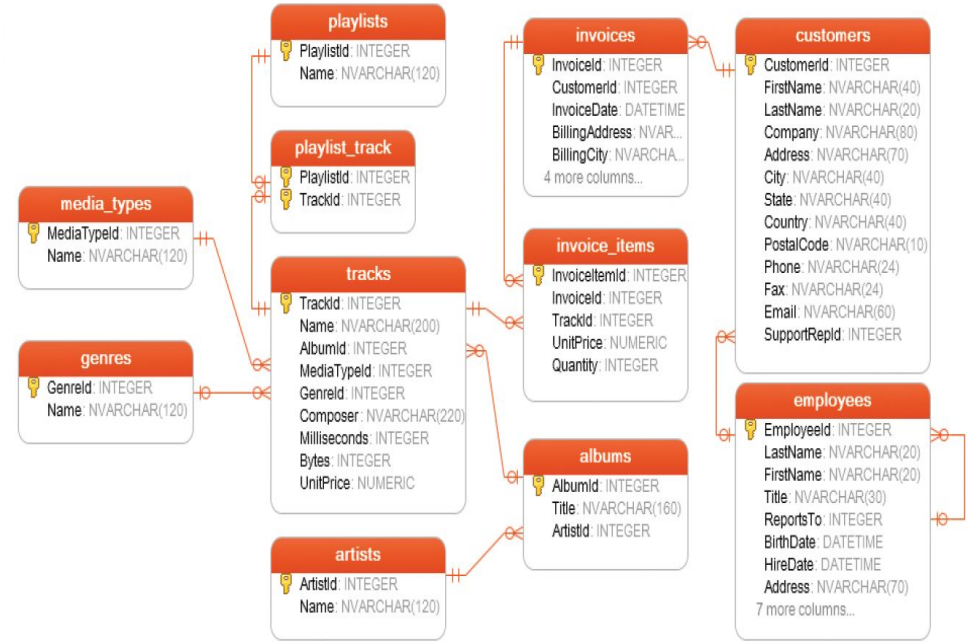
CREATE TABLE-Example

```
CREATE TABLE employee
       (first_name VARCHAR(15),
        last_name VARCHAR(20),
        age INT,
        hire_date DATE);
```



Please add a table to your existing sql_course database: The table name will be **vacation_plan** of the employees for this summer. Column names:

- place_id
- country
- hotel_name
- employee_id
- vacation_length
- budget



DROP TABLE

The DROP TABLE statement is used to drop an existing table in a database.

Syntax:

```
DROP TABLE table_name;
```

```
TRUNCATE TABLE table_name;
```

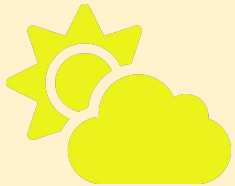
INSERT INTO

INSERT DATA to our vacation_plan table

Syntax:

INSERT INTO table_name (column1, column2 ,..)
VALUES(value1, value2 ,....);

INSERT INTO table1 (column1,column2 ,..)
VALUES
(value1,value2 ,...),
(value1,value2 ,...),
...
(value1,value2 ,...);



Constraints

Constraints are the rules specified for data in a table. We can limit the type of data that will go into a table with the constraints. We can define the constraints with the CREATE TABLE statement or ALTER TABLE statement.

Constraints

Constraints	
Constraint Name	**Definition**
NOT NULL	Ensures that a column cannot have a NULL value
DEFAULT	Sets a default value for a column when no value is specified
UNIQUE	Ensures that all values in a column are different
PRIMARY KEY	Uniquely identifies each row in a table
FOREIGN KEY	Uniquely identifies a row/record in another table

Primary Key

The primary key is a column in our table that makes each row (aka, record) unique.

Syntax

```
CREATE TABLE table_name(
  column_1 INT PRIMARY KEY,
  column_2 TEXT,
  ...
);
```

Primary Key

Syntax (Alternative)

```
1  CREATE TABLE table_name(
2    column_1 INT,
3    column_2 TEXT,
4    ...
5    PRIMARY KEY (column_1)
6  );
```

Foreign Key

Foreign key is a column in a table that uniquely identifies each row of another table. That column refers to a primary key of another table. This creates a kind of link between the tables.

Foreign Key

customers

```sql
CREATE TABLE customers (customer_id INT PRIMARY KEY,
first_name TEXT,
second_name TEXT);
```

orders

```sql
CREATE TABLE orders (
    order_id INT PRIMARY KEY,
    order_number INT,
    customer_id INT,
    FOREIGN KEY (customer_id)
     REFERENCES customers (customer_id)
);
```

Not Null

A column can include NULL values. A NULL value is a special value that means the value is unknown or does not exist.
All columns (except primary key's column) in a table can hold NULL values unless we explicitly specify NOT NULL constraints.

Not Null

Syntax

```
1  CREATE TABLE table_name (
2      column_name type_name NOT NULL,
3    ...);
4
```

Please drop the table as you've just created writing
DROP TABLE vacation_plan;
Then, recreate the vacation_plan table adding constraints as below:
Column names:

- place_id -> PRIMARY KEY
- country
- hotel_name -> NOT NULL
- employee_id -> FOREIGN KEY
- vacation_length
- budget

4 ALTER TABLE

ALTER TABLE

The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.
It is also used to add and drop various constraints on an existing table.

To add a column in a table, use the following syntax:

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

Add a column to your vacation_plan table named “city”.

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

ALTER TABLE

To delete a column in a table, use the following syntax:

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

To change the data type of a column in a table, use the following syntax:

```
ALTER TABLE table_name
MODIFY COLUMN  column_name data_type;
```

Drop the city column from vacation_plan table.

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

THANKS!