



DATABASE SYSTEMS

Dr. Noha Nagy

Lecture 4

SQL: DDL

Reference

2

The book is available

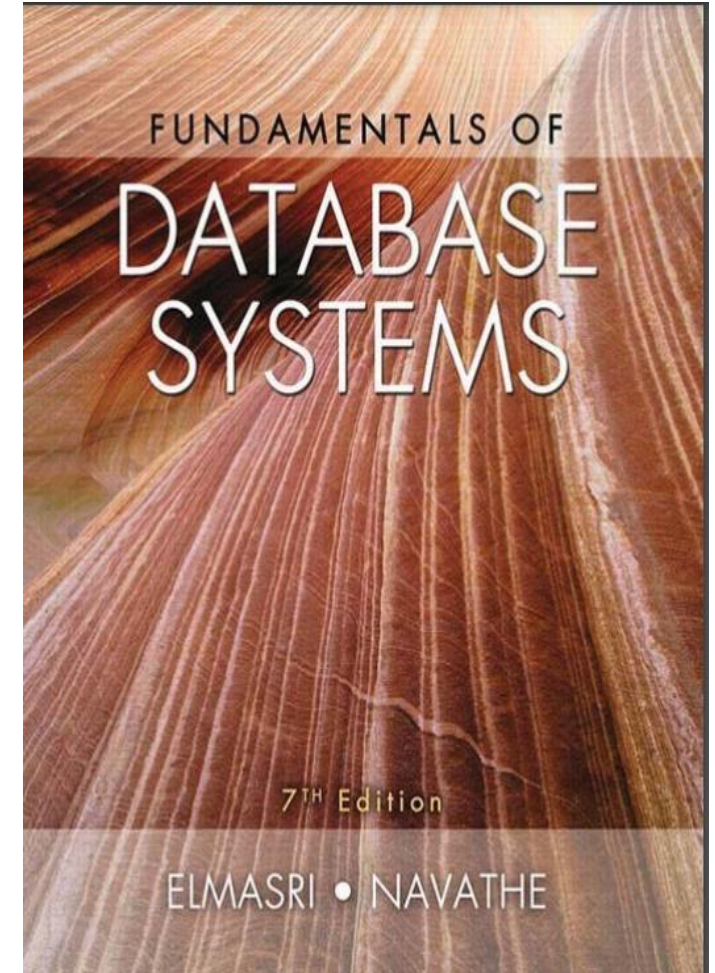
<http://auhd.edu.ye/upfiles/elibrary/Azal2020-01-22-12-28-11-76901.pdf>

Chapter 1 [Introduction]

Chapter 2 [DB System Concepts]

Chapter 5 [Relational Model]

Chapter 6 [SQL]



SQL

3

- *Structured Query language* **SQL** is pronounced as “S-Q-L” or sometimes as “See-Quel” which is the standard language for dealing with **Relational Databases**.
- It is effectively used to create, insert, search, update, delete, modify database records.

DDL: Data Definition Language

4

- It is used to create and modify the structure of database objects in database. Includes statements like:
 - **CREATE:** Creates a new table or any other database object
 - **ALTER:** Modifies an existing database object, such as a table
 - **DROP:** Removes an entire table or any other object in database

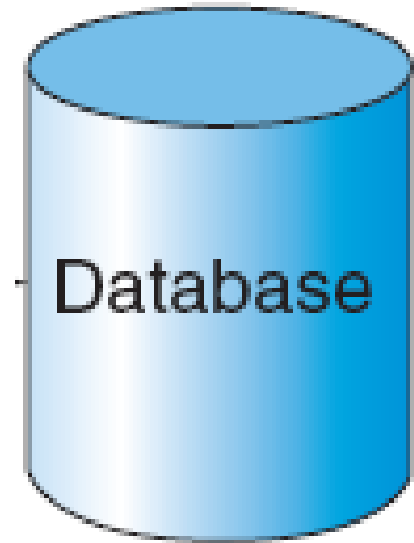
CREATE Statement

5

- Creating a new database:
 - ▣ CREATE DATABASE databasename;
 - ▣ Example:

create database customers;

- Major CREATE statements:
 - ▣ CREATE TABLE—defines a new table and its columns
 - ▣ CREATE VIEW—defines a logical table from one or more tables or views (Not part of the course)



Steps in Table Creation

6

1. Identify data types for **attributes**
2. Identify columns that can and **cannot be null**
3. Identify columns that must be **unique** (candidate keys)
4. Identify **primary key–foreign key** mates
5. Determine **default** values
6. Identify **constraints** on columns (domain specifications)
7. Create the table

Data Types for Attributes

7

- **Common Data Types for table attributes are:**
- **1- Numeric:** integer number (INTEGER, INT, AND SMALLINT), and floating number (FLOAT, REAL, and DOUBLE).
- **2- Character:** data types are either fixed length (CHAR (n), where n is the number of character) or variable length (VARCHAR(n)).
- **3- Boolean:** TRUE or FALSE.
- **4- Timestamp**

Create Table Syntax

8

```
CREATE TABLE <Table Name> (  
  <column1 datatype constraint_1 constraint_2 >,  
  <column2 datatype>,  
  ....  
  constraint_3,  
  Constraint_4);
```


Create Table: Constraints

9

- **NOT NULL:** A Constraint that ensures that a column cannot have NULL value.
- **DEFAULT:** A Constraint that provides a default value for a column when none is specified.
- **UNIQUE :** A Constraint that ensures that all values in a column are different.
- **PRIMARY Key:** A Constraint that uniquely identify each row/record in a database table (NOT NULL + UNIQUE)
- **FOREIGN KEY (FK):** A Constraint that ensures referential integrity. A foreign key from 1 table to another is used link a tuple in the 1st table to a unique tuple in the 2nd table.
- **CHECK:** A constraint that ensures that all values in a column satisfy a certain condition.

Create Table: Example

10

Persons(ID, LastName , FirstName , Age)

SQL Server / Oracle / MS Access:

```
CREATE TABLE Persons (  
    ID int NOT NULL PRIMARY KEY,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int  
);
```

Syntax for SQL Server

11

```
CREATE TABLE Persons (  
    Personid int IDENTITY(1,1) PRIMARY KEY,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int  
);
```

Note:

Identity(seed,increment)

Reserved Words

12

- In SQL, certain words are reserved. These are called keywords or reserved words. These words can not be used as identifiers i.e., as column names in SQL.
- To use these reserved words, write the column names inside the square brackets '[]' so that they are not read as reserved words.


- Example:

- Year

- Registered(SSN, CrsCode, Semester, Year)

Create Table Registered

```
(  
SSN int Foreign Key References Student(SSN),  
CrsCode int Foreign Key References Course(Crscod),  
Semester varchar (45) NOT NULL,  
[Year] varchar (45);  
Primary Key (SSN, CrsCode)  
);
```



Using Check

13

SQL Server / Oracle / MS Access:

```
CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int CHECK (Age>=18)  
);
```

Using Default Value

14

Persons(ID, LastName , FirstName , Age , City)

```
CREATE TABLE Persons (  
    ID int NOT NULL,  
    LastName varchar(255) NOT NULL,  
    FirstName varchar(255),  
    Age int,  
    City varchar(255) DEFAULT 'Cairo'  
);
```

Foreign Key Constraint

15

Persons Table

PersonID	LastName	FirstName	Age
1	Hansen	Ola	30
2	Svendson	Tove	23
3	Pettersen	Kari	20

Orders Table

OrderID	OrderNumber	PersonID
1	77895	3
2	44678	3
3	22456	2
4	24562	1

Order ID is the
primary key

The "PersonID" column in the "Persons" table is the **PRIMARY KEY** in the "Persons" table.

The "PersonID" column in the "Orders" table is a **FOREIGN KEY** in the "Orders" table.

The **FOREIGN KEY** constraint prevents invalid data from being inserted into the foreign key column, because it has to be one of the values contained in the parent table.

Notice that the "PersonID" column in the "Orders" table points to the "PersonID" column in the "Persons" table.

Foreign Key Example

16

SQL Server / Oracle / MS Access:

```
CREATE TABLE Orders (  
    OrderID int NOT NULL PRIMARY KEY,  
    OrderNumber int NOT NULL,  
    PersonID int FOREIGN KEY REFERENCES Persons(PersonID)  
);
```


SQL ALTER TABLE: Add Column

17

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

```
ALTER TABLE table_name  
ADD column_name datatype;
```

The following SQL adds an "Email" column to the "Customers" table:

Example

```
ALTER TABLE Customers  
ADD Email varchar(255);
```

Changing Tables

18

<u>Student ID</u>	Student Name	Faculty	Department	Age
Numeric(5)	Varchar(100)	Varchar(30)	Varchar(30)	Integer

□ ALTER TABLE changes column specifications:

▣ ALTER TABLE Student ADD (Type Varchar(2))

<u>Student ID</u>	Student Name	Faculty	Department	Age	Type
Numeric(5)	Varchar(100)	Varchar(30)	Varchar(30)	Integer	Varchar(2)

□ ALTER TABLE Student DROP Age;

OR

□ ALTER TABLE Student DROP COLUMN Age;

<u>Student ID</u>	Student Name	Faculty	Department	Type
Numeric(5)	Varchar(100)	Varchar(30)	Varchar(30)	Varchar(2)

ALTER TABLE: Constraints

19

Adding and dropping Constraints

ALTER TABLE<table name>

ADD CONSTRAINT < CONSTRAINT Syntax>

ALTER TABLE<table name>

ADD CONSTRAINT < CONSTRAINT Name > < CONSTRAINT Syntax>

ALTER TABLE<table name>

Drop CONSTRAINT < CONSTRAINT Name >

ALTER TABLE: Constraints Example

20

```
ALTER TABLE Customers  
ADD PRIMARY KEY (SSN, Telephone);
```

Or

```
ALTER TABLE Registered  
ADD CONSTRAINT PK1 PRIMARY KEY (SSN, Telephone );
```

-- constraint name is unique across database

Alter Table : Examples

21

```
ALTER TABLE STUDENT
```

```
ADD CONSTRAINT UQ_Std_Name unique (Name);
```

```
ALTER TABLE STUDENT
```

```
DROP CONSTRAINT UQ_Std_Name;
```

```
ALTER TABLE DEPARTMENT
```

```
ADD CONSTRAINT CK_DEPTCODE CHECK (DEPTCODE IN ('IS','CS','IT',  
'DS'))
```

Alter Table : Examples

22

```
ALTER TABLE DEPARTMENT  
DROP CONSTRAINT CK_DEPTCODE;
```

```
ALTER TABLE Student  
ALTER COLUMN City SET DEFAULT 'Cairo';
```

```
ALTER TABLE Persons  
ALTER COLUMN City DROP DEFAULT;
```

SQL DROP : For deleting Tables

□ DROP TABLE statement allows you to remove tables from your schema:

□ DROP TABLE TableName

SQL DROP : For deleting databases

- DROP Database statement allows you to delete all the database:
 - ▣ DROP database <database name>;
 - ▣ **Example:** DROP database mydatabase;

Questions

25

1. Create a table for employee with the following columns:
 - ▣ Employee_id (int)
 - ▣ Employee_name (varchar(50))
 - ▣ Employee_address (varchar(250))
 - ▣ Employee_age (int)
2. Drop the table

Answer

26

```
CREATE TABLE employees (  
Employee_id INT NOT NULL,  
Employee_name VARCHAR(50) NOT NULL,  
Employee_address VARCHAR(250) NOT NULL,  
Employee_age INT NOT NULL,  
PRIMARY KEY (Employee_id));
```

```
DROP Table employees;
```

Update Operations on Relations

27

- ❑ INSERT or add a tuple.
- ❑ DELETE a tuple.
- ❑ UPDATE a tuple.
- ❑ Integrity constraints **should not be violated** by the update operations.
- ❑ Updates may **propagate** to cause other updates automatically. This may be necessary to maintain integrity constraints.

Possible Violations for Delete Operation

28

- DELETE may violate only referential integrity:
 - ▣ If the primary key value of the tuple being deleted is referenced from other tuples in the database
 - Can be remedied by several actions: RESTRICT, CASCADE, SET NULL
 - **RESTRICT** option: reject the deletion
 - **CASCADE** option: propagate the new primary key value into the foreign keys of the referencing tuples
 - **SET NULL** option: set the foreign keys of the referencing tuples to NULL
 - ▣ One of the above options must be specified during database design for each foreign key constraint

Foreign Key Constraint

29

Student

<u>SID</u>	SName	City	DID
111	Ahmed	Cairo	1
112	Ali	Cairo	2
113	Osman	Giza	1
114	Nabila	Giza	4
115	Hoda	Giza	4

Department

<u>DeptID</u>	DName	Head of Dept
1	Information Systems	Mohamed Nour
2	Information Technology	Osama
3	Operational Research	Eyad
4	Computer Science	Ahmed

Foreign Key Constraint

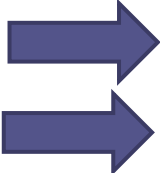
Restrict → Can't be deleted there are student in this department

Set Null → FK Set Null

Cascade → Delete all student in this department

30

Student



<u>SID</u>	SName	City	<u>DID</u>
111	Ahmed	Cairo	1
112	Ali	Cairo	2
113	Osman	Giza	1
114	Nabila	Giza	4
115	Hoda	Giza	4

Department

<u>DeptID</u>	DName	Head of Dept
1	Information Systems	Mohamed Nour
2	Information Technology	Osama
3	Operational Research	Eyad
4	Computer Science	Ahmed

```
CREATE TABLE Orders (  
    OrderID int NOT NULL,  
    OrderNumber int NOT NULL,  
    PersonID int,  
    PRIMARY KEY (OrderID),  
    CONSTRAINT FK_PersonOrder FOREIGN KEY (PersonID)  
    REFERENCES Persons(PersonID)  
);
```

You can also add how the Foreign key updates will be handled:

```
REFERENCES Persons(ID)  
ON DELETE CASCADE  
ON UPDATE CASCADE  
ON DELETE SET NULL);
```

Foreign Key Constraint: Restrict

32

Restrict → Can't be deleted there are student in this department

Student

<u>SID</u>	SName	City	DID
111	Ahmed	Cairo	1
112	Ali	Cairo	2
113	Osman	Giza	1
114	Nabila	Giza	4
115	Hoda	Giza	4

Delete department 4

Error: Can't delete this department
delete all the students related to this
department first

Department

<u>DeptID</u>	DName	Head of Dept
1	Information Systems	Mohamed Nour
2	Information Technology	Osama
3	Operational Research	Eyad
4	Computer Science	Ahmed

Foreign Key Constraint: Cascade

33

Cascade → Delete all student in this department

Student

<u>SID</u>	SName	City	DID
111	Ahmed	Cairo	1
112	Ali	Cairo	2
113	Osman	Giza	1
114	Nabila	Giza	4
115	Hoda	Giza	4

Delete department 4

Department

<u>DeptID</u>	DName	Head of Dept
1	Information Systems	Mohamed Nour
2	Information Technology	Osama
3	Operational Research	Eyad
4	Computer Science	Ahmed

Foreign Key Constraint: Set Null

34

Set Null → FK Set NULL

Delete Department 4

Student

<u>SID</u>	SName	City	DID
111	Ahmed	Cairo	1
112	Ali	Cairo	2
113	Osman	Giza	1
114	Nabila	Giza	
115	Hoda	Giza	

Department

<u>DeptID</u>	DName	Head of Dept
1	Information Systems	Mohamed Nour
2	Information Technology	Osama
3	Operational Research	Eyad
4	Computer Science	Ahmed

Foreign Key Constraint: Cascade

35

Student

<u>SID</u>	SName	City	DID
111	Ahmed	Cairo	1
112	Ali	Cairo	2
113	Osman	Giza	1
114	Nabila	Giza	4
115	Hoda	Giza	4

**Update department
Set DeptID= 5 Where DeptID= 4**

Department

<u>DeptID</u>	DName	Head of Dept
1	Information Systems	Mohamed Nour
2	Information Technology	Osama
3	Operational Research	Eyad
4	Computer Science	Ahmed

Foreign Key Constraint: Cascade

36

Student

<u>SID</u>	SName	City	DID
111	Ahmed	Cairo	1
112	Ali	Cairo	2
113	Osman	Giza	1
114	Nabila	Giza	5
115	Hoda	Giza	5

**Update department
Set DeptID= 5 Where DeptID= 4**

Department

<u>DeptID</u>	DName	Head of Dept
1	Information Systems	Mohamed Nour
2	Information Technology	Osama
3	Operational Research	Eyad
5	Computer Science	Ahmed

Alter Table : Examples

37

```
ALTER TABLE STUDENT
```

```
ADD CONSTRAINT FK_1 FOREIGN KEY (Major) REFERENCES  
Department (DeptCode) ON DELETE SET NULL ON UPDATE  
CASCADE;
```