



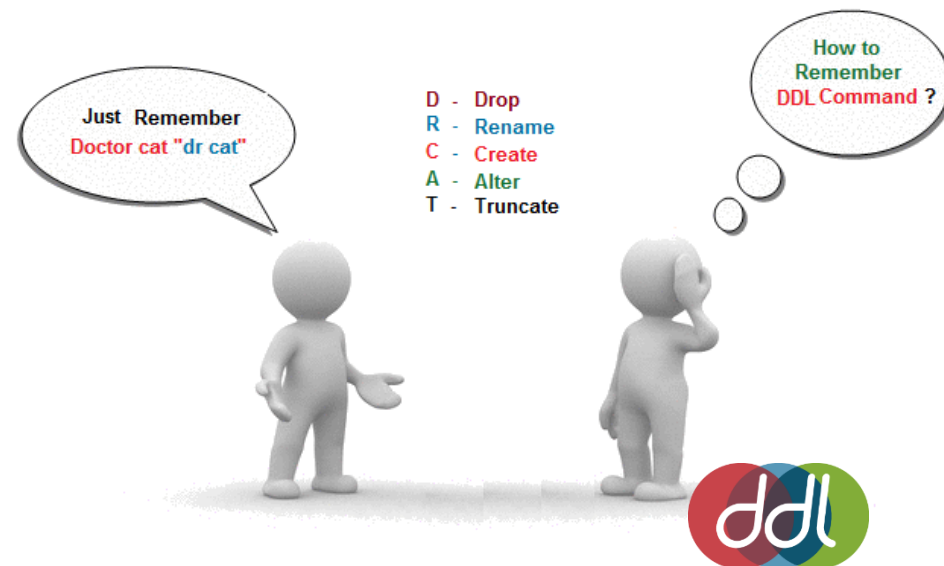
DDL Commands

Môn học: Hệ quản trị cơ sở dữ liệu [*Buổi 2*]

GV: Nguyễn Mai Huy

DDL

Data Definition Language

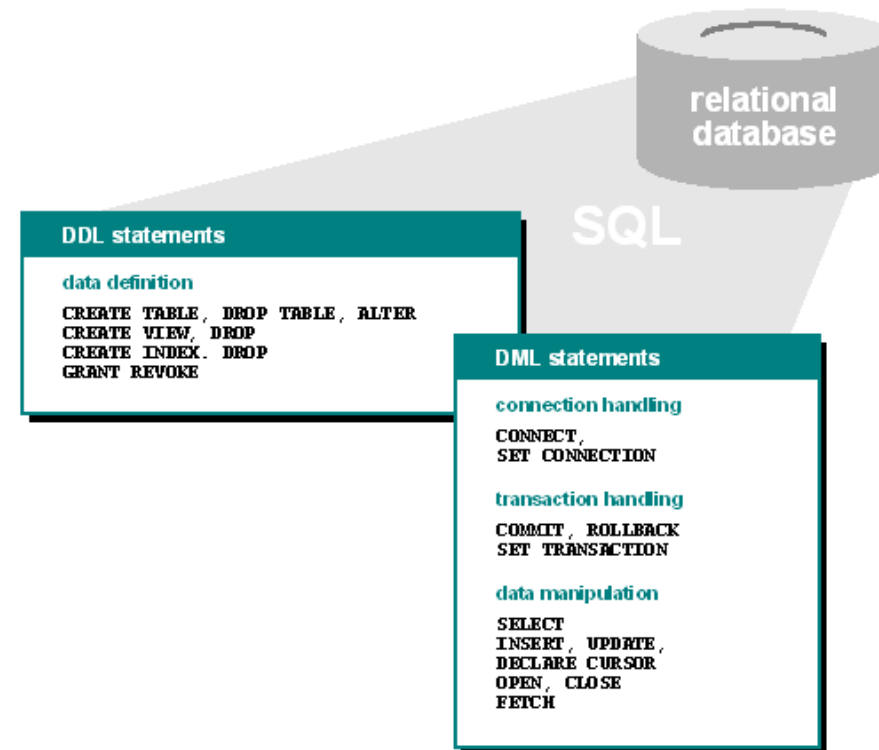


Data Definition Language

Các lệnh **DDL** trong **SQL** được sử dụng để tạo lược đồ cơ sở dữ liệu nhằm xác định loại và cấu trúc của dữ liệu sẽ được lưu trữ trong CSDL.

Các lệnh **DDL** trong **T-SQL** được chia thành 4 loại chính như sau:

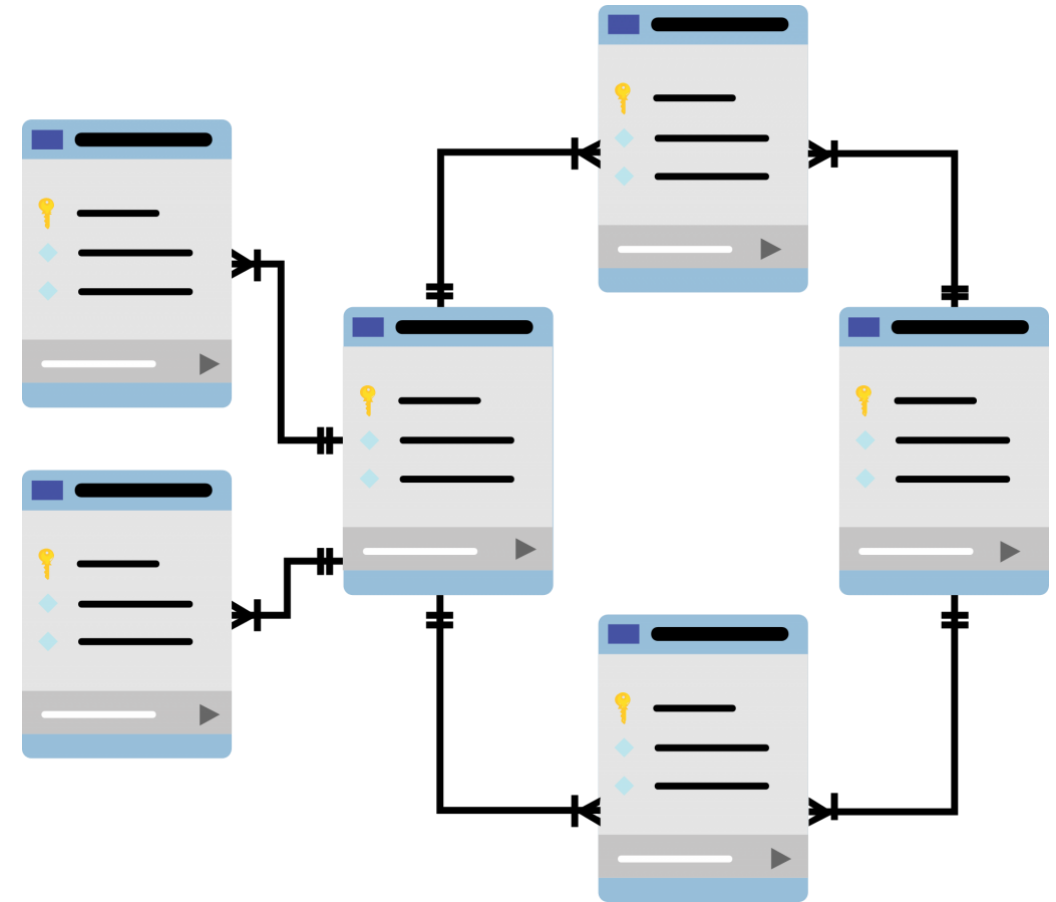
- ❖ CREATE
- ❖ ALTER
- ❖ DROP
- ❖ TRUNCATE



DDL & Database object

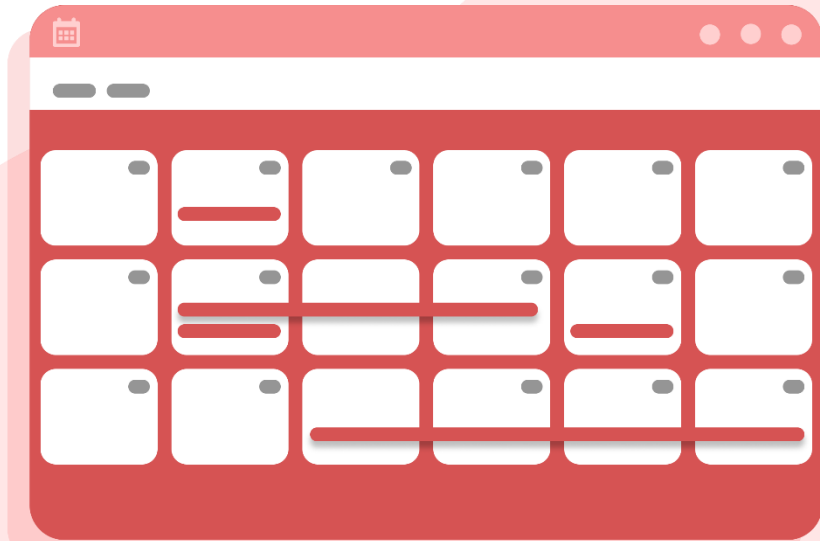
- ❖ **Create Database** <Database_Name>;
- ❖ **Alter Database** <Db_OldName>
Modify name = <Db_NewName>;
- ❖ **Drop Database** <DbName>;

Create Database OnlineTrading;



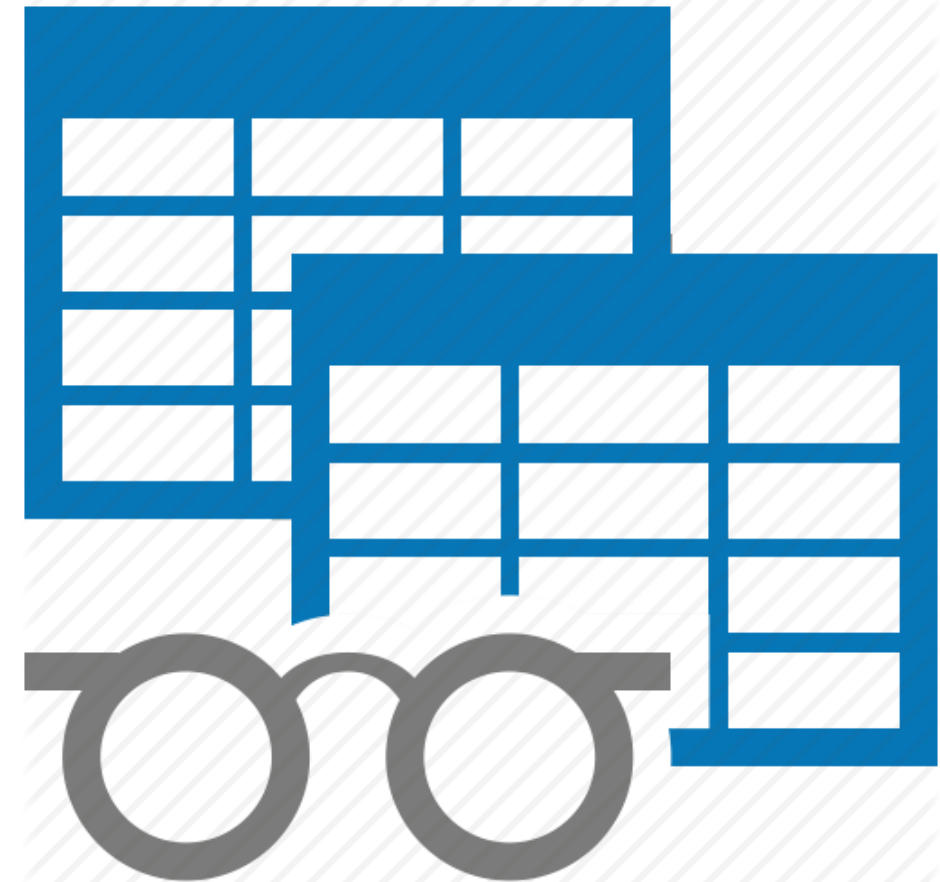
DDL & Table

- ❖ **Create Table** <Table_Name>
- ❖ **Alter Table** <Table_Name>
- ❖ **Drop Table** <Table_Name>
- ❖ **Truncate Table** <Table_Name>



DDL & **View**

- ❖ **Create View** <View_Name>
- ❖ **Alter View** <View_OldName>
- ❖ **Drop View** <ViewName>



DDL & Trigger

- ❖ **Create Trigger** <Trigger_Name>
- ❖ **Alter Trigger** <Trigger_Name>
- ❖ **Drop Trigger** <TriggerName>



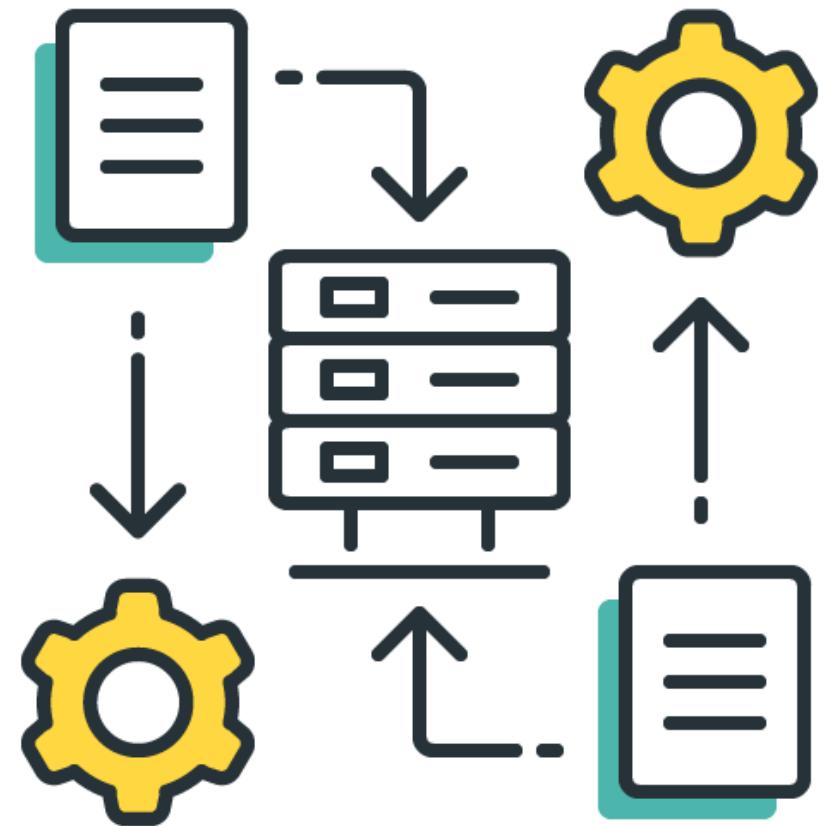
DDL & **Function**

- ❖ **Create Function** <Function_Name>
- ❖ **Alter Function** <Function_Name>
- ❖ **Drop Function** <FunctionName>

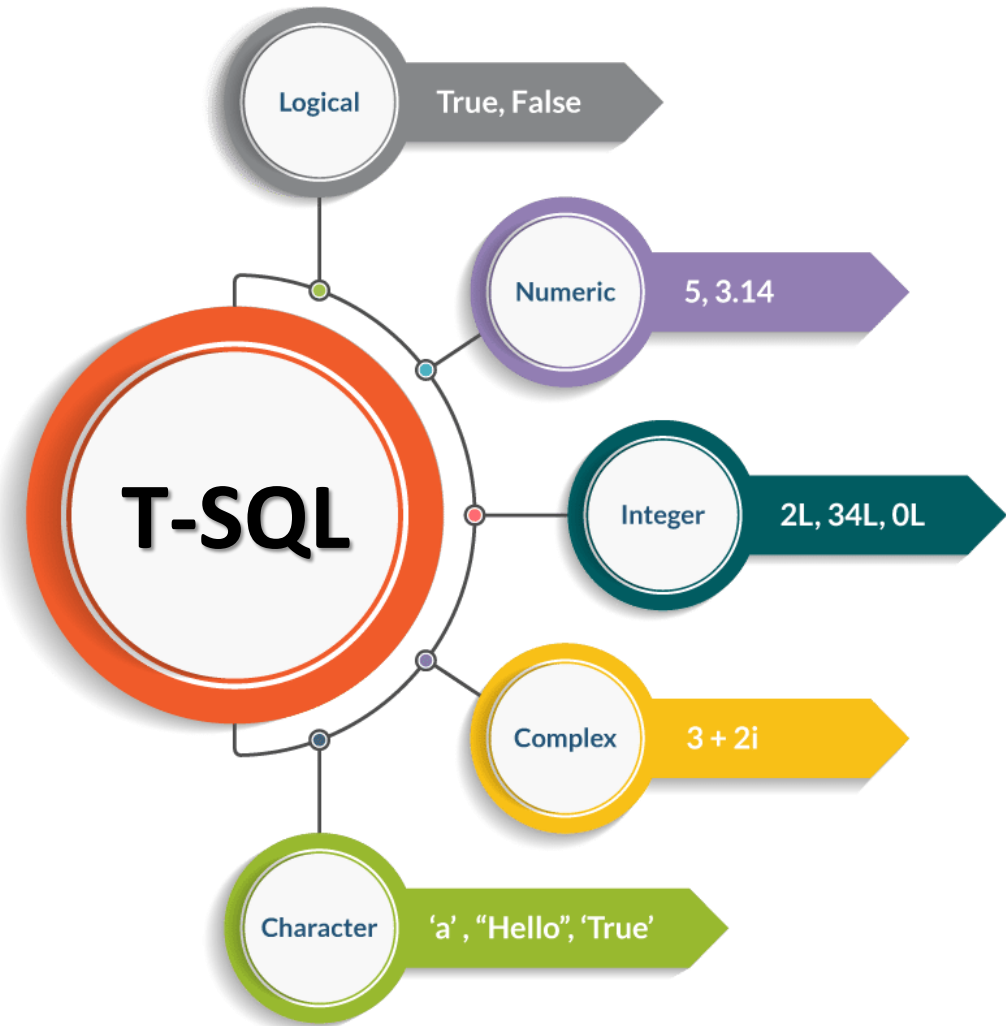
$f(x)$

DDL & Stored Procedure

- ❖ **Create Procedure** <Procedure_Name>
- ❖ **Alter Procedure** <Procedure_Name>
- ❖ **Drop Procedure** <ProcedureName>



Data types



Data types :: T-SQL

The Transact SQL language allow you to use various data types like

- ❖ **Binary Strings**: Binary, Varbinary
- ❖ **Character Strings**: Char, Varchar, Text
- ❖ **Date and Time**: Date, Datetime, Datetime2, Datetimeoffset, Smalldatetime, Time
- ❖ **Numerics**: Bigint, Int, Smallint, Tinyint, Decimal, Numeric, Bit, Float, Real, ...
- ❖ **Unicode Character Strings**: Nchar, Nvarchar, NText
- ❖ **Other Data Types**: Uniqueidentifier, Table, Cursor, SQL_VARIANT, XML, ...

String data types

Data type	Description	Max size	Storage
char(n)	Fixed width character string	8,000 characters	Defined width
varchar(n)	Variable width character string	8,000 characters	2 bytes + number of chars
varchar(max)	Variable width character string	1,073,741,824 characters	2 bytes + number of chars
text	Variable width character string	2GB of text data	4 bytes + number of chars
nchar	Fixed width Unicode string	4,000 characters	Defined width x 2
nvarchar	Variable width Unicode string	4,000 characters	
nvarchar(max)	Variable width Unicode string	536,870,912 characters	
ntext	Variable width Unicode string	2GB of text data	
binary(n)	Fixed width binary string	8,000 bytes	
varbinary	Variable width binary string	8,000 bytes	
varbinary(max)	Variable width binary string	2GB	
image	Variable width binary string	2GB	

Numeric data types

Data type	Description	Storage
bit	Integer that can be 0, 1, or NULL	
tinyint	Allows whole numbers from 0 to 255	1 byte
smallint	Allows whole numbers between -32,768 and 32,767	2 bytes
int	Allows whole numbers between -2,147,483,648 and 2,147,483,647	4 bytes
bigint	Allows whole numbers between -9,223,372,036,854,775,808 and 9,223,372,036,854,775,807	8 bytes
decimal(p,s)	<p>Fixed precision and scale numbers.</p> <p>Allows numbers from $-10^{38} + 1$ to $10^{38} - 1$.</p> <p>The p parameter indicates the maximum total number of digits that can be stored (both to the left and to the right of the decimal point). p must be a value from 1 to 38. Default is 18.</p> <p>The s parameter indicates the maximum number of digits stored to the right of the decimal point. s must be a value from 0 to p. Default value is 0</p>	5-17 bytes

Numeric data types

Data type	Description	Storage
numeric(p,s)	<p>Fixed precision and scale numbers.</p> <p>Allows numbers from $-10^{38} + 1$ to $10^{38} - 1$.</p> <p>The p parameter indicates the maximum total number of digits that can be stored (both to the left and to the right of the decimal point). p must be a value from 1 to 38. Default is 18.</p> <p>The s parameter indicates the maximum number of digits stored to the right of the decimal point. s must be a value from 0 to p. Default value is 0</p>	5-17 bytes
smallmoney	Monetary data from -214,748.3648 to 214,748.3647	4 bytes
money	Monetary data from -922,337,203,685,477.5808 to 922,337,203,685,477.5807	8 bytes
float(n)	<p>Floating precision number data from $-1.79E + 308$ to $1.79E + 308$.</p> <p>The n parameter indicates whether the field should hold 4 or 8 bytes. float(24) holds a 4-byte field and float(53) holds an 8-byte field. Default value of n is 53.</p>	4 or 8 bytes
real	Floating precision number data from $-3.40E + 38$ to $3.40E + 38$	4 bytes

Date and Time data types

Data type	Description	Storage
datetime	From January 1, 1753 to December 31, 9999 with an accuracy of 3.33 milliseconds	8 bytes
datetime2	From January 1, 0001 to December 31, 9999 with an accuracy of 100 nanoseconds	6-8 bytes
smalldatetime	From January 1, 1900 to June 6, 2079 with an accuracy of 1 minute	4 bytes
date	Store a date only. From January 1, 0001 to December 31, 9999	3 bytes
time	Store a time only to an accuracy of 100 nanoseconds	3-5 bytes
datetimeoffset	The same as datetime2 with the addition of a time zone offset	8-10 bytes
timestamp	Stores a unique number that gets updated every time a row gets created or modified. The timestamp value is based upon an internal clock and does not correspond to real time. Each table may have only one timestamp variable	

Other data types

Data type	Description
sql_variant	Stores up to 8,000 bytes of data of various data types, except text, ntext, and timestamp
uniqueidentifier	Stores a globally unique identifier (GUID)
xml	Stores XML formatted data. Maximum 2GB
cursor	Stores a reference to a cursor used for database operations
table	Stores a result-set for later processing

T-SQL Operators

❖ Arithmetic Operators

(add, subtract, multiply, divide, Modulo: **+, -, *, /, %**)

❖ Bitwise Operators

(and, or, exclusive or: **&, /, ^**)

❖ Comparison Operators

(equal, greater than, less than, not equal: **=, >, <, >=, <=, <>**)

❖ Compound Operators

(add equals, subtract equals, multiply equals: **+=, -=, *=, /=, %=, &=, |=, /=**)

❖ Logical Operators

(**all, any, some, in, between, and, or, not, like, exists**)

Wildcard Characters in T-SQL

Symbol	Description	Example
%	Represents zero or more characters	bl% finds bl, black, blue, and blob
_	Represents a single character	h_t finds hot, hat, and hit
[]	Represents any single character within the brackets	h[oa]t finds hot and hat, but not hit
^	Represents any character not in the brackets	h[^oa]t finds hit, but not hot and hat
-	Represents a range of characters	c[a-b]t finds cat and cbt



Wildcard Characters :: Example

LIKE Operator	Description
WHERE CustomerName LIKE 'a%'	Finds any values that starts with "a"
WHERE CustomerName LIKE '%a'	Finds any values that ends with "a"
WHERE CustomerName LIKE '%or%'	Finds any values that have "or" in any position
WHERE CustomerName LIKE '_r%'	Finds any values that have "r" in the second position
WHERE CustomerName LIKE 'a_%_ %'	Finds any values that starts with "a" and are at least 3 characters in length
WHERE ContactName LIKE 'a%o'	Finds any values that starts with "a" and ends with "o"



DDL & Table



DLL & Table

❖ **Create Table** <Table_Name>;

❖ **Alter Table**

✓ **Alter Table** <Table_Name> **Add column** <col_name> <data_type>;

✓ **Alter Table** <Table_Name> **Drop column** <col_name>;

✓ **Alter Table** <Table_Name> **Alter column** <col_name> <data_type>;

❖ **Drop Table** <TbName>;

❖ **Truncate Table** <TbName>;

SQL Constraints

- SQL constraints are used to specify rules for the data in a table.
- Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted.
- Constraints can be column level or table level. Column level constraints apply to a column, and table level constraints apply to the whole table.

SQL Constraints

- ✓ **NOT NULL** - Ensures that a column cannot have a NULL value
- ✓ **UNIQUE** - Ensures that all values in a column are different
- ✓ **PRIMARY KEY** - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
- ✓ **FOREIGN KEY** - Uniquely identifies a row/record in another table
- ✓ **CHECK** - Ensures that all values in a column satisfies a specific condition
- ✓ **DEFAULT** - Sets a default value for a column when no value is specified
- ✓ **INDEX** - Used to create and retrieve data from the database very quickly



DDL & Constraint

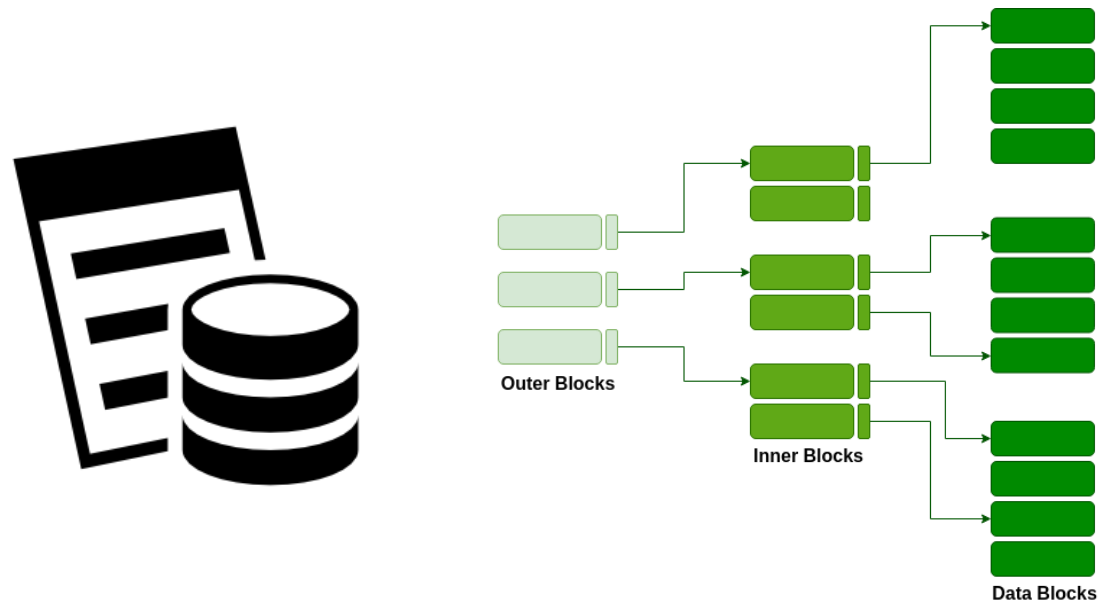
- ✓ Add constraint

Alter Table <table_name> **Add constraint** <constraint_name>

- ✓ Drop constraint

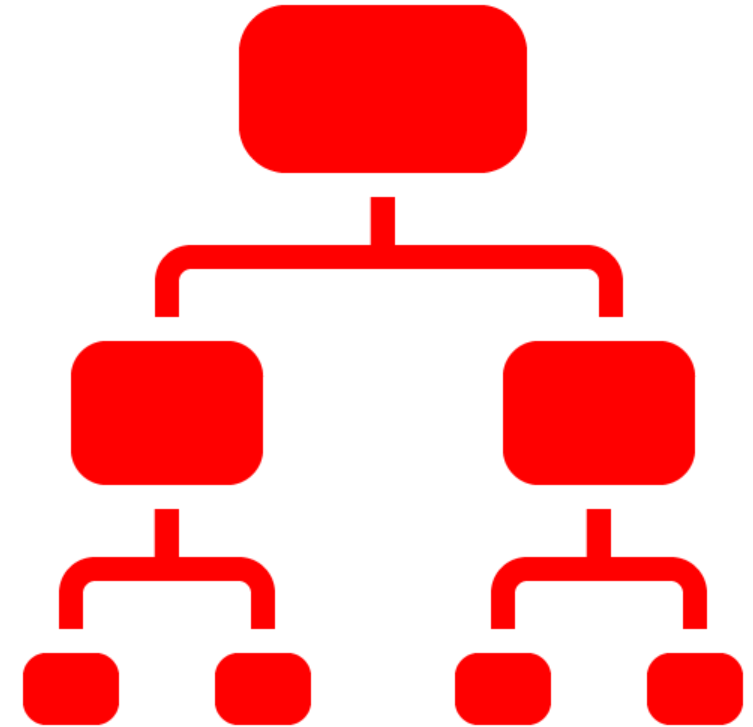
Alter Table <table_name> **Drop constraint** <constraint_name>

Index

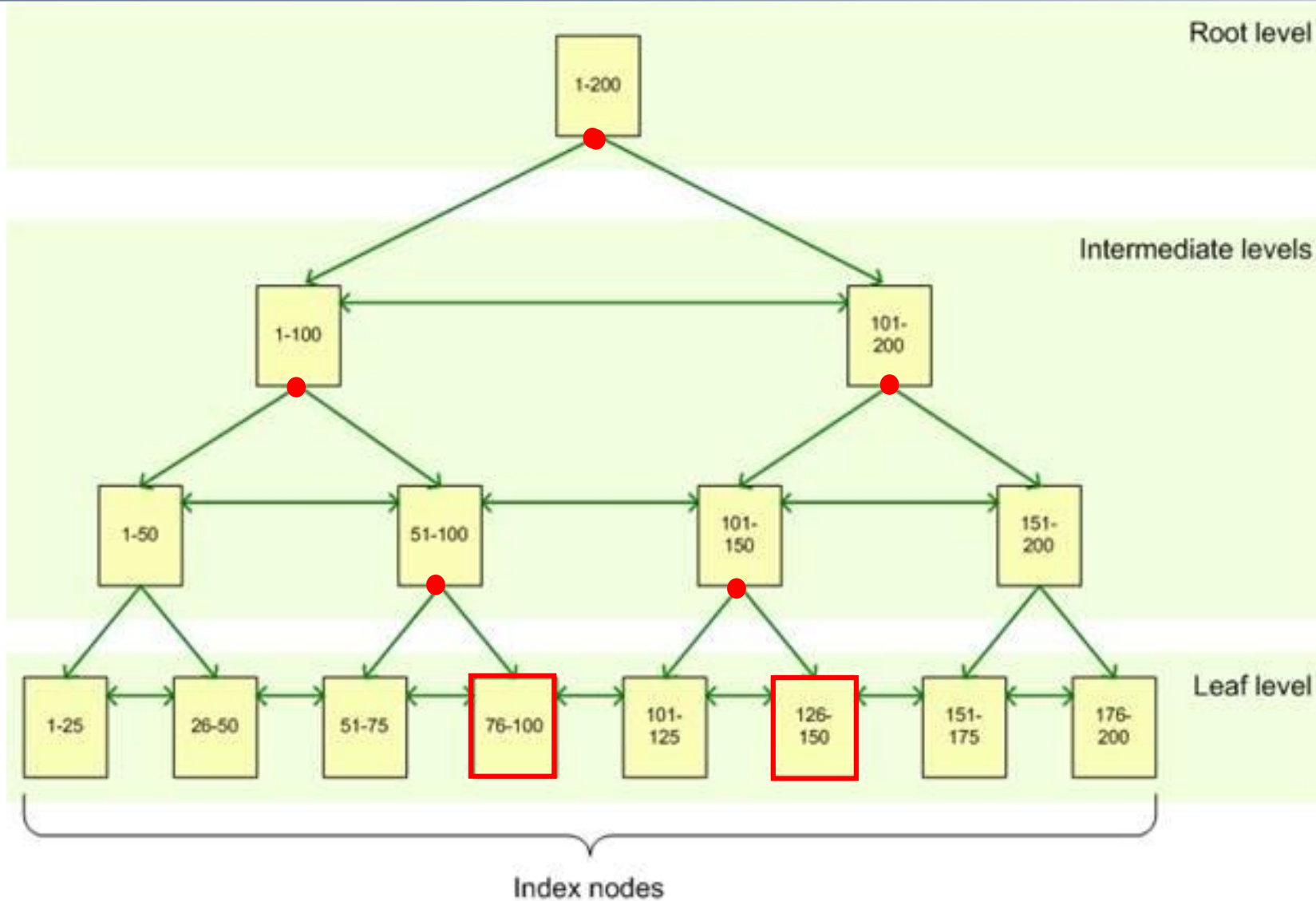


What is index ?

- Index là một ***cấu trúc liên kết*** giữa việc lưu trữ trên ***đĩa với dữ liệu*** của một **Table** hay **View** nhằm tăng tốc độ khi truy xuất thông tin trên các đối tượng dạng này
- Mặc nhiên, Index được tạo ra cho Table/View dựa vào **Primary key**. Những Key này được lưu trữ trong một cấu trúc B-Tree để cho phép SQL Server tìm *Row* hay *liên kết các Rows* với *key tương ứng một cách nhanh nhất*

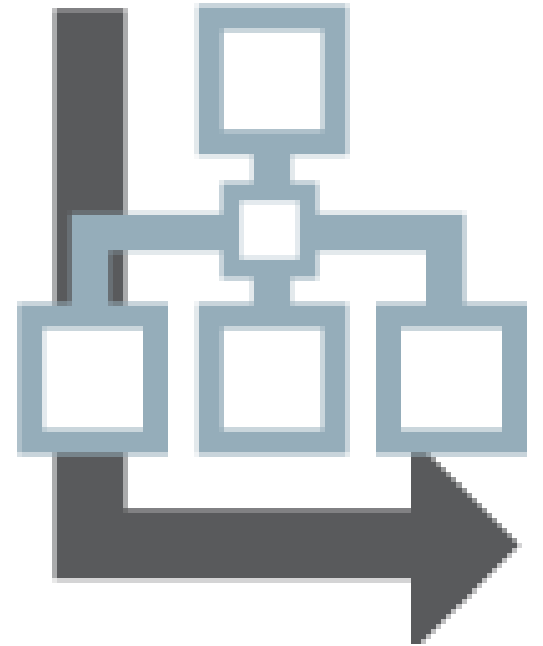


Pages – index nodes



Types of **indexes**

- ❖ **Clustered indexes:** *dữ liệu sẽ được sắp xếp trên bảng theo giá trị của field tương ứng. Một bảng chỉ có thể có 1 field được index theo dạng này. Mặc nhiên trong một table, primary key field sẽ được index theo dạng này*
- ❖ **Nonclustered Indexes:** *Một bảng tham chiếu được tạo ra tương ứng với các Index được tạo ở dạng này, bao gồm 2 cột. Cột thứ nhất chứa các giá trị của field được index và cột này được sắp xếp theo trật tự quy ước (do người lập trình thiết lập khi tạo Noncluster index); cột còn lại sẽ chứa địa chỉ, trỏ tới dòng có chứa giá trị tương ứng ở trên bảng.*

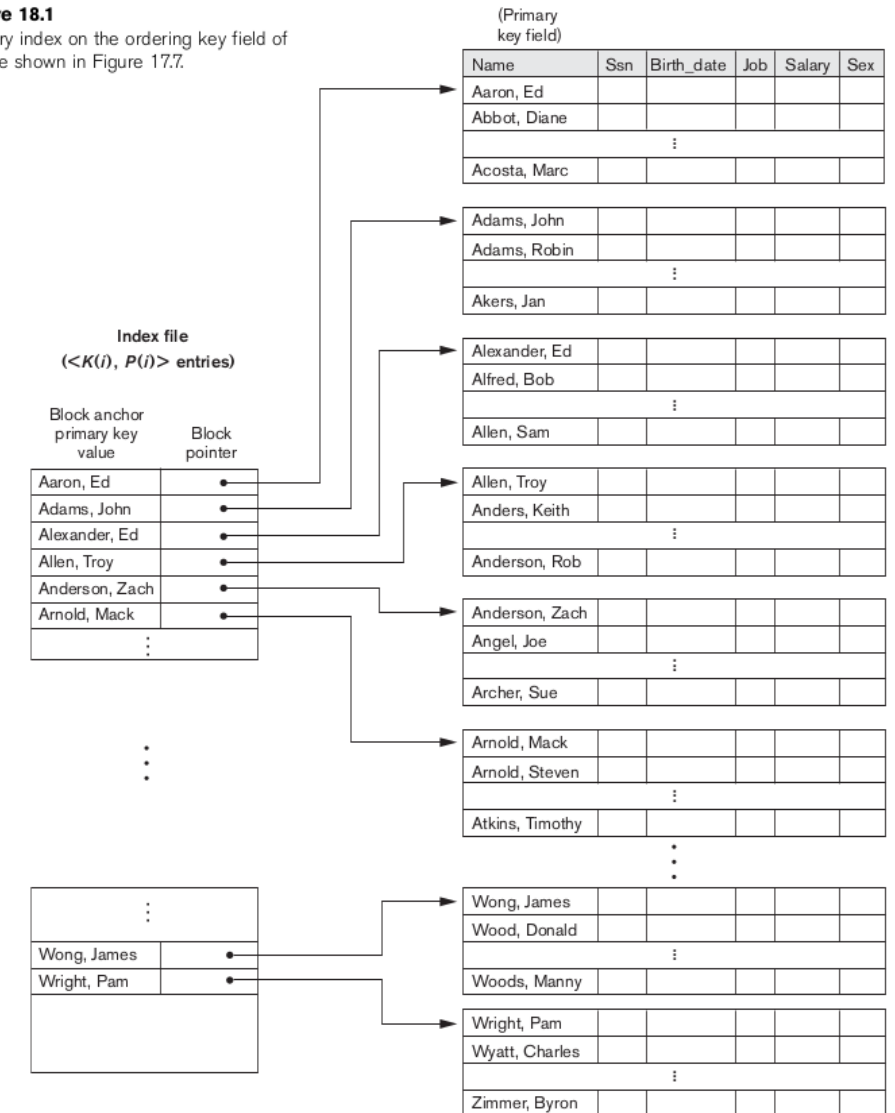


Clustered indexes

- ❖ **Clustered index** sắp xếp và lưu các dòng dữ liệu ở trong bảng dựa trên khóa. Bao gồm những cột được định nghĩa trong Index.
- ❖ Mỗi bảng chỉ có thể có 1 **Clustered Index**, bởi dữ liệu của bảng chỉ có thể sắp xếp lưu trữ theo một cách duy nhất.
- ❖ Các dòng dữ liệu của một bảng được lưu theo trật tự sắp xếp khi bảng đó được tạo **Clustered index**
- ❖ Một bảng chứa **Clustered index** thì được gọi là **Clustered table**
- ❖ Nếu một Table không chứa **Clustered index** thì các dòng dữ liệu trong nó được lưu trữ theo cấu trúc không sắp xếp gọi là Heap.

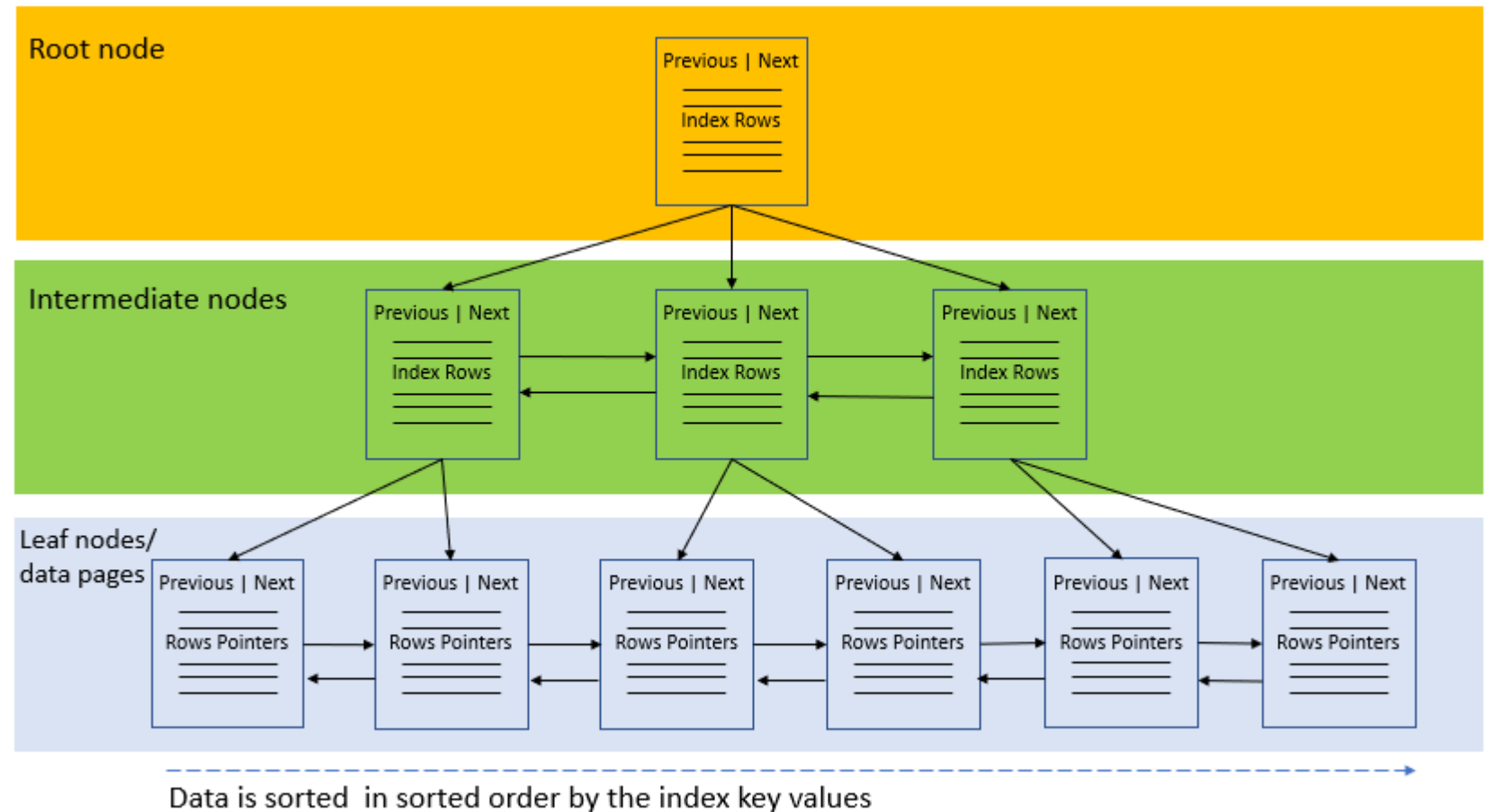
Figure 18.1

Primary index on the ordering key field of the file shown in Figure 17.7.



Nonclustered indexes

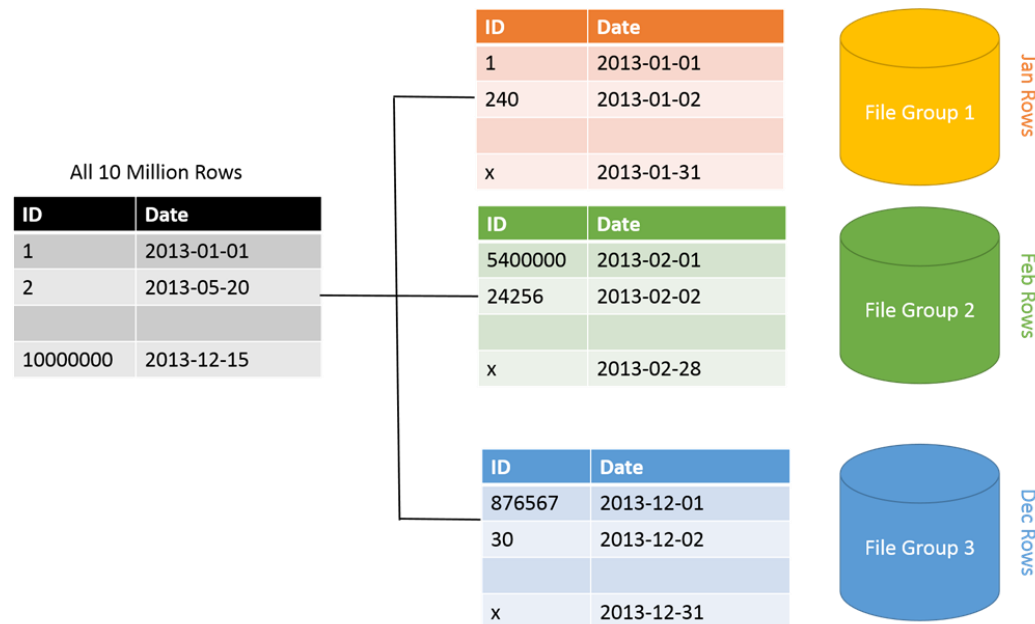
- ❖ **Nonclustered index** là một cấu trúc không đồng nhất dựa trên các dòng dữ liệu.
- ❖ Một **Nonclustered index** chứa các giá trị khóa rời rạc, và các giá trị khóa này là một con trỏ, trỏ tới dòng chứa giá trị tương ứng



Create | Drop Index

- **Create** [**Clustered** | **Nonclustered**] **Index** <index_name> **ON** <object> (column1 [Asc|Desc] , column2 [Asc|Desc], ...)
- **Drop Index** <table_name.index_name>

Syntax for Create – Alter table



Create table

```
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  
);
```

Example : Create table

-- 2: Tạo Table [Customers] chứa Thông tin khách hàng -----

```
create table kháchHang
(
    maKH varchar(10) primary key not null,
    tenKH nvarchar(50) not null,
    soDT varchar(20) ,
    email varchar(50),
    diaChi nvarchar(250),
    ngaySinh datetime ,
    gioiTinh bit default 1,
    ghiChu ntext
)
go
```

Example : Create table

```
-- 3: Tạo Table [Articles] chứa thông tin về các bài viết phục vụ cho quảng bá sản phẩm, ---  
-- xu hướng mua sắm hiện nay của người tiêu dùng , ... -----  
create table baiViet  
(  
    maBV varchar(10) primary key not null,  
    tenBV nvarchar(250) not null,  
    hinhDD varchar(max),  
    ndTomTat nvarchar(2000),  
    ngayDang datetime ,  
    loaiTin nvarchar(30),  
    noiDung nvarchar(4000),  
    taiKhoan varchar(20) not null ,  
    daDuyet bit default 0,  
    foreign key (taiKhoan) references taiKhoanTV(taiKhoan)  
)  
go
```

Example : Create table

```
-- 6: Tạo Table [OrderDetails] chứa thông tin chi tiết của các đơn hàng ---  
--      mà khách đã đặt mua với các mặt hàng cùng số lượng đã chọn -----  
create table ctDonHang  
(  
    soDH varchar(10) not null foreign key references donHang(soDH),  
    maSP varchar(10) not null foreign key references sanPham(maSP),  
    soLuong int,  
    giaBan bigint,  
    giamGia BIGINT,  
    PRIMARY KEY (soDH, maSP)  
)  
go
```


Alter table – Add Column

```
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;
```

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

Alter table – Alter Column

```
ALTER TABLE table_name  
ALTER COLUMN column_name datatype;
```

```
ALTER TABLE Persons  
ALTER COLUMN DateOfBirth year;
```

Alter table – Drop Column

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

```
ALTER TABLE Customers  
DROP COLUMN Email;
```

Alter table – Add | Drop constraint

```
ALTER TABLE Persons ADD CONSTRAINT  
    <constraint_Name> <constraint_type> (column | expression);
```

```
ALTER TABLE Persons  
ADD CONSTRAINT PK_Person PRIMARY KEY (ID, LastName);
```

```
ALTER TABLE Persons DROP CONSTRAINT PK_Person;
```

Nhớ gì ?!!!

- DDL: Create – Alter – Drop <Db_Object>
- Data types & Operator
- DDL & Table
- Create table & Constraint
- Alter table :: Add, Drop : Column, Constraint
- Clustered Index & Nonclustered Index

Tài liệu tham khảo

- Itzik Ben-Gan, “**Microsoft® SQL Server ® 2012 T-SQL Fundamentals**”, O’Reilly Media Inc, 2012
- Itzik Ben-Gan, Dejan Sarka, Ed Katibah, Greg Low, Roger Wolter, and Isaac Kunen, “**Inside Microsoft SQL Server 2008: T-SQL Programming**”, Microsoft Press, 2010
- w3schools, “**Introduction to SQL**”,
https://www.w3schools.com/sql/sql_intro.asp, 10:54PM, 18/06/2020
- Microsoft SQL Server Tutorial, “**Tutorials for SQL Server**”,
<https://docs.microsoft.com/en-us/sql/sql-server/tutorials-for-sql-server-2016>,
10:54 PM, 18/06/2020