



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

SQL Server Part DDL and DML

آزمایشگاه پایگاه داده

حمیدرضا رمضانی

h.ramezany72@gmail.com

اشیاء اصلی در SQL SERVER

- Table
- Constraint
- View
- Index
- Stored Procedure
- Function
- Trigger
- User-defined Data Type
- Rule
- And

آزمایشگاه پایگاه داده

حمیدرضا رمضانی



جدول

مهمترین شیی در بانک اطلاعاتی است
از هر جدول (حتی الامکان) برای نگهداشت یک موجودیت استفاده می شود.
در نامگذاری جداول قواعد زیر سودمند است :

- نام گذاری کامل به جای نام گذاری مخفف .
- نام گذاری مفرد به جای جمع .
- حرف ابتدایی کلمات اصلی بزرگ نوشته شود.
- فیلد کلید اصلی حتی الامکان دارای نام **Id** باشد.
- نام کلید فرعی حتی الامکان از نام جدول پدر به همراه **Id_** یا **Id** استفاده شود .

قيود (CONSTRAINT)

بايد ها و نبايدهاى يك جدول در راستاى جامعيت داده ها
(Data Integrity)

انواع :

- 1) Unique Constraint
- 2) Primary Key Constraint
- 3) Foreign Key Constraint
- 4) Check Constraint

آزمایشگاه پایگاه داده
حمیدرضا رمضانی

ایجاد جدول

- CREATE TABLE table_name

```
(  
    column_name1      data_type      ,  
    column_name2      data_type      ,  
    column_name3      data_type      ,  
    ....  
)
```

- Example:

- CREATE TABLE Persons

```
(  
    P_Id int,  
    LastName varchar(255),  
    FirstName varchar(255),  
    Address varchar(255),  
    City varchar(255)  
)
```

P_Id	LastName	FirstName	Address	City

قیود SQL

- NOT NULL
- UNIQUE
- PRIMARY KEY
- FOREIGN KEY
- CHECK
- DEFAULT

آزمایشگاه پایگاه داده

حمیدرضا رمضانی



قید NOT NULL

• برای Null نبودن یک ویژگی باید به صورت زیر عمل کرد:

```
CREATE TABLE Persons  
(  
  P_Id int NOT NULL,  
  LastName varchar(255) NOT NULL,  
  FirstName varchar(255),  
  Address varchar(255),  
  City varchar(255)  
)
```



Space

قید UNIQUE

• برای unique بودن یک ویژگی باید به صورت زیر عمل کرد:

```
CREATE TABLE Persons  
(  
  P_Id int NOT NULL,  
  LastName varchar(255) NOT NULL,  
  FirstName varchar(255),  
  Address varchar(255),  
  City varchar(255),  
  CONSTRAINT uc_PersonID UNIQUE (P_Id,LastName)  
)
```


PRIMARY KEY قيد

• برای primary key بودن یک ویژگی باید به صورت زیر عمل کرد:

- CREATE TABLE Persons
(
P_Id int NOT NULL PRIMARY KEY,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Address varchar(255),
City varchar(255)
)

PRIMARY KEY قید

- CREATE TABLE Persons
(
P_Id int NOT NULL,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Address varchar(255),
City varchar(255),
CONSTRAINT pk_PersonID PRIMARY KEY
(P_Id,LastName)
)

قید FOREIGN KEY

In Parent Table	Status	In Child Table
Select	Nothing	Nothing
Insert	Nothing	Check If exists in Parent
Update	No Action	Return Error
	Cascade	Update Foreign Key in Child Rows
	Set Null	Sets Null Value To Foreign Key if Nullable Or Return Error if Not Nullable
	Set Default	Set Default Value if Exists in Parent Table Or Return Error if Not Exists.
Delete	No Action	Return Error
	Cascade	Delete Child Rows
	Set Null	Sets Null Value To Foreign Key if Nullable Or Return Error if Not Nullable
	Set Default	Set Default Value if Exists in Parent Table Or Return Error if Not Exists.

هنگام **Insert / Update** در جدول فرزند :
 در این هنگام مقادیر وارد شده برای ستون کلید خارجی در جدول فرزند ، با مقادیر موجود در کلید اصلی جدول پدر مقایسه شده ، در صورت عدم وجود این مقدار در جدول پدر، اجازه عملیات صادر نمی شود .

هنگام **Delete / Update** در جدول پدر :
 در این هنگام امکان انتخاب یکی از حالت‌های زیر در جدول فرزند وجود دارد :

- No Action (1)
- Cascade (2)
- Set Null (3)
- Set Default (4)

FOREIGN KEY قيد

#1:

```
CREATE TABLE Orders
(  
  O_Id int NOT NULL PRIMARY KEY,  
  OrderNo int NOT NULL,  
  P_Id int FOREIGN KEY REFERENCES  
  Persons(P_Id)
```

#2:

```
CREATE TABLE Orders
(  
  O_Id int NOT NULL,  
  OrderNo int NOT NULL,  
  P_Id int,  
  PRIMARY KEY (O_Id),  
  CONSTRAINT fk_PerOrders FOREIGN KEY (P_Id) REFERENCES Persons(P_Id)  
)
```

قید CHECK

```
CREATE TABLE Persons
(
  P_Id int NOT NULL,
  LastName varchar(255) NOT NULL,
  FirstName varchar(255),
  Address varchar(255),
  City varchar(255),
  CONSTRAINT chk_Person CHECK (P_Id>0 AND City='Sandnes')
)
```

فید DEFAULT

```
CREATE TABLE Persons  
(  
  P_Id int NOT NULL,  
  LastName varchar(255) NOT NULL,  
  FirstName varchar(255),  
  Address varchar(255),  
  City varchar(255) DEFAULT 'Tehran'  
)
```

تغییر تعریف جدول (حذف یا اضافه کردن فیلد)

Alter table table_name

Add new_column DataType Nullability

مثال:

```
alter table reserve  
add sailor_name int not null;
```

```
alter table reserve  
drop column sailor_name;
```

تغییر تعریف جدول

- تغییر نوع فیلد

Alter table table_name

Alter Column column_name New_DataType [Nullability]

- اضافه کردن محدودیت

Alter table table_name

Add CONSTRAINT constraint_name CHECK (check_syntax)

- حذف محدودیت:

Alter table table_name

Drop Constraint constraint_name

اضافه کردن اطلاعات به پایگاه داده

create table *instructor* (

<i>ID</i>	char(5),
<i>name</i>	varchar(20) not null,
<i>dept_name</i>	varchar(20),
<i>salary</i>	numeric(8,2))

- **insert into** *instructor* **values** ('10211', 'Smith', 'Biology', 66000);
- **insert into** SalesStaff2 (StaffGUID, FirstName, LastName) **values** (NEWID(), 'Stephen', 'Jiang');



اضافه کردن اطلاعات به پایگاه داده

create table *instructor* (

<i>ID</i>	char(5),
<i>name</i>	varchar(20) not null,
<i>dept_name</i>	varchar(20),
<i>salary</i>	numeric(8,2))

- **insert into** *instructor* **values** ('10211', 'Smith', 'Biology', 66000);
- **insert into** SalesStaff2 (StaffGUID, FirstName, LastName) **values** (NEWID(), 'Stephen', 'Jiang');



دستور DELETE و DROP

- این دستور برای حذف یک یا چند ردیف جدول استفاده می شود. شکل کلی آن به صورت زیر است:

```
DELETE [FROM] table_or_view_name  
WHERE search_condition
```

```
DELETE FROM tblSailor  
WHERE sailor_rank<5
```

- DROP TABLE table_name

بخش DML دستور SELECT

SELECT expressions
FROM tables
[WHERE conditions];

دستور کامل:

SELECT [ALL | DISTINCT]
[TOP (top_value) [PERCENT] [WITH TIES]]
expressions
FROM tables
[WHERE conditions]
[GROUP BY expressions]
[HAVING condition]
[ORDER BY expression [ASC | DESC]];

بخش DML دستور SELECT

ALL

Optional. Returns all matching rows.

DISTINCT

Optional. Removes duplicates from the result set. Learn more about the [DISTINCT clause](#)

TOP (top_value)

Optional. If specified, it will return the top number of rows in the result set based on *top_value*. For example, TOP(10) would return the top 10 rows from the full result set.

PERCENT

Optional. If specified, then the top rows are based on a percentage of the total result set (as specified by the *top_value*). For example, TOP(10) PERCENT would return the top 10% of the full result set.

بخش DML دستور SELECT

WITH TIES

Optional. If specified, then rows tied in last place within the limited result set are returned. This may result in more rows be returned than the *TOP* parameter permits.

GROUP BY expressions

Optional. It collects data across multiple records and groups the results by one or more columns.

HAVING condition

Optional. It is used in combination with the GROUP BY to restrict the groups of returned rows to only those whose the condition is TRUE.

ORDER BY expression

Optional. It is used to sort the records in your result set. ASC sorts in ascending order and DESC sorts in descending order.

بخش DML دستور SELECT

مثال:

```
SELECT *  
FROM inventory  
WHERE quantity > 5 ORDER BY inventory_id ASC;
```

مثال:

```
SELECT inventory_id, inventory_type, quantity  
FROM inventory  
WHERE inventory_id >= 555 AND inventory_type =  
'Software'  
ORDER BY quantity DESC, inventory_id ASC;
```