

آزمایش سوم

◀ توجه کنید: بخش‌هایی که با این علامت مشخص شده‌اند کارهایی است که باید در آزمایشگاه انجام و تحویل دهید.

تغییر اطلاعات

برای تغییر دادن اطلاعات در جداول پایگاه داده از دستوره‌های INSERT، UPDATE و DELETE می‌توان استفاده کرد. در ادامه‌ی مطلب درباره‌ی هریک از این دستورها به‌صورت مفصل‌تر صحبت می‌کنیم.

دستور INSERT

از این دستور برای افزودن یک ردیف جدید به یک جدول استفاده می‌شود. شکل کلی این دستور به‌صورت زیر است:

```
INSERT [INTO] <table_or_view_name> [(column_list)]  
VALUES ({DEFAULT | NULL | expression} [, ...n])
```

در صورتی که نخواهیم برای تمام فیلدها یا ستون‌ها مقداری مشخص کنیم، لیست این فیلدها را در قسمت column_list مشخص می‌کنیم. مقداری که باید در آن ردیف قرار گیرد به‌صورت لیستی از مقادیر که با کاما از هم جدا شده‌اند، بعد از عبارت VALUES ظاهر می‌شوند. هر مقدار می‌تواند به‌صورت یک عبارت، NULL یا DEFAULT باشد.

مثال:

```
INSERT INTO USER1 (uid, uname)  
VALUES ('u001', 'ali');
```

به‌جای مشخص کردن مقادیر یک ردیف به‌صورت صریح مانند مثال بالا می‌توان نتایج یک پرس‌وجو را در یک جدول ذخیره کرد. در این حالت شکل دستور به‌صورت زیر است:

```
INSERT [INTO] <table_or_view_name> [(column_list)]  
SELECT ...
```

به‌عنوان مثال می‌خواهیم اطلاعات مربوط به پیغام‌های کاربری خاص را در جدولی موقت ذخیره کنیم:

```
CREATE TABLE MESSAGE_TEMP  
(  
    id INT,  
    title VARCHAR(50),  
    body VARCHAR(1000)  
);
```

```
INSERT INTO MESSAGE_TEMP (id, title, body)
```

```
SELECT mid, title, body
FROM MESSAGE1
WHERE creatorId='u001';
```

فیلدهای از نوع IDENTITY: در هر جدول حداکثر یک فیلد می‌تواند از نوع identity باشد، به این معنی که SQL Server منحصر به فرد بودن آن را خود تضمین می‌کند و در واقع هرگاه ردیفی در جدول اضافه شود، مقدار جدیدی به صورت اتوماتیک در آن فیلد قرار می‌گیرد. بنابراین اگر سعی کنید در دستور INSERT برای این فیلدها مقداری به صورت صریح مشخص کنید با پیام خطایی روبه‌رو می‌شوید.

به عنوان مثال اگر دستورهای زیر را اجرا کنیم:

```
CREATE TABLE MESSAGE2 (
    mid INT IDENTITY(1, 1) PRIMARY KEY,
    title NVARCHAR(100) NOT NULL,
    body NVARCHAR(500) );
```

```
INSERT INTO MESSAGE2 (mid, title, body)
VALUES (2, 'title1', 'body1');
```

با پیام خطای زیر روبه‌رو می‌شویم:

```
Cannot insert explicit value for identity column in table 'MESSAGE2' when
IDENTITY_INSERT is set to OFF.
```

اگر بخواهیم مقدار صریحی برای این نوع فیلد تعریف کنیم باید از دستور SET IDENTITY_INSERT ON استفاده کنیم:

```
SET IDENTITY_INSERT MESSAGE2 ON;

INSERT INTO MESSAGE2 (mid, title, body)
VALUES (2, 'title1', 'body1');
```

دستور UPDATE

از این دستور برای تغییر مقدار(های) یک یا چند ردیف یک جدول استفاده می‌شود. شکل کلی آن به صورت زیر است:

```
UPDATE <table_or_view_name>
SET column_name = {expression | DEFAULT | NULL} [ , ...n ]
WHERE <search_condition>
```

مثال:

```
UPDATE MESSAGE1
SET body = '(from user 001)'+body
WHERE creatorId='u001';
```

قالب دیگر این دستور به صورت زیر است:

```
UPDATE <table_or_view_name>
SET column_name = {expression | DEFAULT | NULL} [ , ...n ]
FROM <table_source>
WHERE <search_condition>
```

قسمت FROM معمولا برای مشخص کردن JOIN دو یا چند جدول استفاده می‌شود. به عنوان مثال می‌خواهیم عنوان تمام پیام‌هایی را که کاربری بنام Ali آنها را ارسال کرده به "I am Ali" تغییر دهیم:

```
UPDATE MESSAGES
SET MESSAGES.title = 'I am Ali'
FROM MESSAGES INNER JOIN Users ON MESSAGES.senderId=users.uid
WHERE users.FirstName = 'I am Ali'
```

دستور DELETE

از این دستور برای حذف یک یا چند ردیف یک جدول استفاده می‌شود. شکل کلی آن به صورت زیر است:

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

به عنوان مثال می‌خواهیم inbox یک کاربر خاص (با کد ۱۰) را حذف کنیم:

```
DELETE FROM MESSAGE1
WHERE receiverId = 10
```

در مثال زیر اطلاعات تمام سفارش‌هایی که دو سال از ثبت آنها گذشته است به جدولی دیگر منتقل و از جدول جاری حذف می‌شوند. ابتدا تعریف اولیه‌ی جدول سفارش (ORDER1):

```
CREATE TABLE ORDER1 (
    orderNumber INT IDENTITY(1000,1) PRIMARY KEY,
    orderDate DATETIME,
    customerId INT);
```

سپس انتقال اطلاعات به جدولی دیگر:

```
SELECT *
INTO PREV_ORDER
FROM ORDER1
WHERE DATEDIFF(year, orderDate, GETDATE())>1
```

در پایان حذف این سفارش‌ها:

```
DELETE FROM ORDER1
WHERE DATEDIFF(year, orderDate, GETDATE())>1
```

اگر بخواهید تمام ردیف‌های یک جدول را با سرعت بیشتری نسبت به دستور DELETE حذف کنید می‌توانید از دستور TRUNCATE استفاده کنید. دلیل سرعت بیشتر این دستور آن است که میزان log برای این دستور حداقل است. مثال:

```
TRUNCATE TABLE PREV_ORDER;
```

استفاده از عبارت OUTPUT

ساختار این دستور که به همراه دستورهای تغییر اطلاعات به کار می‌رود به شکل زیر است:

```
INSERT | DELETE | UPDATE ...
```

```
OUTPUT {DELETED | INSERTED | from_table_name} . { * | column_name }
[,...n ] INTO { @table_variable | output_table } [WHERE ...]
```

دو جدول مجازی با نام‌های DELETED و INSERTED وجود دارند که حاوی مقادیر اصلی و پس از تغییر جدول به‌روز شده‌اند. این دو جدول دارای ساختاری مشابه با جدول اصلی هستند و بنابراین می‌توان تمام یا بعضی از فیلدهای آن‌ها را انتخاب کرد تا در جواب حاضر باشند. به مثال زیر دقت کنید. جدول EMPLOYEE به‌صورت زیر تعریف شده است:

```
CREATE TABLE EMPLOYEE (
    eid INT IDENTITY(100, 1) PRIMARY KEY,
    firstName NVARCHAR(50) NOT NULL,
    lastName NVARCHAR(50) NOT NULL,
    departmentNumber INT NOT NULL,
    salary INT);
```

فرض کنید که مقادیر زیر در این جدول وجود دارد (شکل ۱). می‌خواهیم حقوق کارکنان دپارتمان ۱ را افزایش دهیم و سپس لیستی از حقوق قبلی و فعلی آن‌ها داشته باشیم:

eid	firstName	lastName	departmentNu...	salary
100	ali	alavi	1	300000
101	reza	rezai	2	350000
102	reyhane	samani	1	320000
103	maryam	salamat	1	NULL

شکل ۱: مقادیر اولیه‌ی جدول EMPLOYEE

```
DECLARE @temp TABLE (
    fname NVARCHAR(50),
    lname NVARCHAR(50),
    dNo INT,
    prevSalary INT,
    newSalary INT
);

UPDATE EMPLOYEE
SET salary = salary * 1.2
OUTPUT INSERTED.firstName, INSERTED.lastName,
INSERTED.departmentNumber,
DELETED.salary, INSERTED.salary
INTO @temp
WHERE departmentNumber=1 AND salary IS NOT NULL;

SELECT * FROM @temp;
```

در این حالت خروجی‌ای مشابه شکل ۲ خواهیم داشت.

	fname	lname	d...	prevSalary	newSalary
1	ali	alavi	1	300000	360000
2	reyhane	samani	1	320000	384000

شکل ۲: نتیجه‌ی عبارت OUTPUT

◀ با استفاده از جداول طراحی شده برای سیستم پیامها برای هریک از عملیات زیر دستور مناسبی بنویسید.

۱. دستور *insert* یک رکورد در جدول کاربران با مقادیر دلخواه
۲. دستور حذف کاربر شماره ۱۰
۳. دستور تغییر نام و نام خانوادگی کاربر شماره ۱۰ به مقادیر دلخواه جدید
۴. دستور حذف کاربری که تا کنون هیچ پیامی ارسال نکرده است (کد او بعنوان فرستنده در جدول پیامها یافت نشود)
۵. دستور حذف منطقی پیامی با شماره ۱۲۵