**بسمه تعالی**



برنامه نویسی بانک اطلاعاتی SQL Server

**نویسنده : محمد حسین فخرآوری**

**Data Manipulation Language**

دستوراتی جهت درج ، بروزرسانی و حذف اطلاعات در دیتابیس

**INSERT**

جهت درج رکورد/ رکوردهای جدید در جدول

|  |
| --- |
| INSERT INTO table\_name VALUES (value1,value2,value3,...) |

در این فرمت تنها مقدیر جهت درج رکورد جدید ذکر میشوند و در آن مقادیر بترتیب ستونهای جدول در فیلد های رکورد جدید درج میشوند. درج چند رکورد با یک insert into

|  |
| --- |
| INSERT INTO table\_name  VALUES (value1,value2,value3,...), (value1,value2,value3,...), (value1,value2,value3,...) |

در فرمت ذیل نام ستونها و مقادیر ذکر میشوند

|  |
| --- |
| INSERT INTO table\_name (column1,column2,column3,...) VALUES (value1,value2,value3,...) |

* اگر بخواهیم ستونی مقدار default خود را داشته باشد. در لیست مقادیر کلمه default را بجای آن میگذاریم .

**UPDATE**

جهت بروزرسانی رکورد های موجود در جدول.

|  |
| --- |
| UPDATE table\_name  SET column1=value1,column2=value2,... WHERE some\_column=some\_value |

**SELECT INTO**

فرض کنید می خواهیم بر اساس نتیجه یک select یک جدول بسازیم ؟

با این دستور مطابق با فیلد های موجود در select جدولی ایجاد شده و اطلاعات رکوردهای منتج از select نیز در آن کپی میشود.

|  |
| --- |
| SELECT column\_name(s) INTO newtable FROM table1 |

New table: نام جدول جدیدی است که ساخته میشود  
Table1: جدول اصلی ما که select بروی آن انجام میشود   
نام ستونهایی که در جدول جدید ایجاد شوند و اطلاعات در آنها کپی شود. طبیعتا می توان از \* هنگامی که می خواهیم همه ستونها در جدول جدید ایجاد شوند می توان استفاده کرد  
نکته : جدول جدید با ستونهای همنام با جدول اصلی ساخته خواهد شد. اگر می خواهیم نام ستونهای جدید تغییر کند می توانیم از alias برای ستونهای مورد نظر استفاده کنیم.

**DELETE**

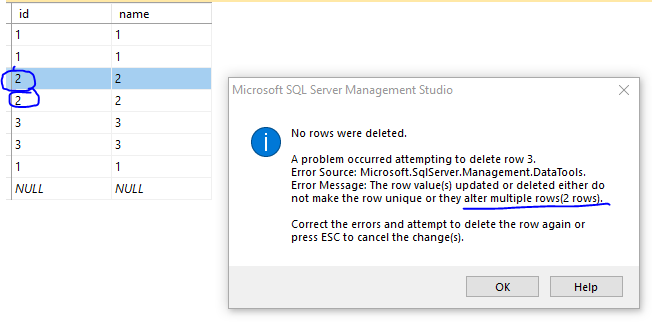
جهت حذف رکوردها از جدول

|  |
| --- |
| DELETE FROM table\_name WHERE some\_column=some\_value |

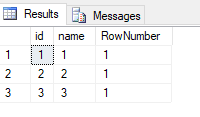
* در صورت عدم ذکر عبارت where تمامی رکورد های جدول حذف خواهند شد. لذا در اجرای دستور delete بایستی نهایت دقت را نمود .

**مشکل حذف داده هایی که جدول آن ها کلید ندارد**

|  |
| --- |
| CREATE TABLE [dbo].[HyperStudents]([id] [int] NULL,[name] [nvarchar](50) NULL) ON [PRIMARY]  GO  INSERT [dbo].[HyperStudents] ([id], [name]) VALUES (1, N'1')  GO  INSERT [dbo].[HyperStudents] ([id], [name]) VALUES (1, N'1')  GO  INSERT [dbo].[HyperStudents] ([id], [name]) VALUES (2, N'2')  GO  INSERT [dbo].[HyperStudents] ([id], [name]) VALUES (2, N'2')  GO  INSERT [dbo].[HyperStudents] ([id], [name]) VALUES (3, N'3')  GO  INSERT [dbo].[HyperStudents] ([id], [name]) VALUES (3, N'3')  GO  INSERT [dbo].[HyperStudents] ([id], [name]) VALUES (1, N'1')  GO |



|  |
| --- |
| WITH HS as  (  SELECT \*, ROW\_NUMBER() over (PARTITION BY id ORDER BY id) as RowNumber  FROM HyperStudents  )  SELECT \* FROM HS where RowNumber > 1  --Delete from HS where RowNumber > 1 |

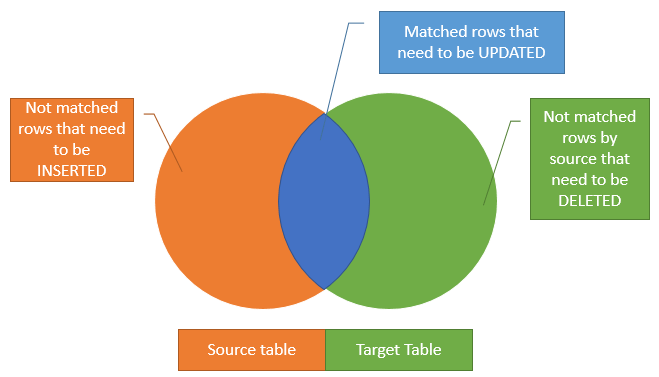


**MERGE**

در واقع این دستور ترکیبی از UPDATE و DELETE و INSERT می باشد. این تصمیم که آیا عملیات بروز رسانی یا درج سطر جدید در جدول مقصد صورت گیرد، همگی بستگی به شرطی دارد که در عبارت ON مشخص می شود.

|  |
| --- |
| MERGE target\_table USING source\_table  ON merge\_condition  WHEN MATCHED  THEN update\_statement  WHEN NOT MATCHED  THEN insert\_statement  WHEN NOT MATCHED BY SOURCE  THEN DELETE; |

* در پایان دستور از semi-colon استفاده شود.
* در این دستور (دیتای source بر روی target) اعمال می شود.

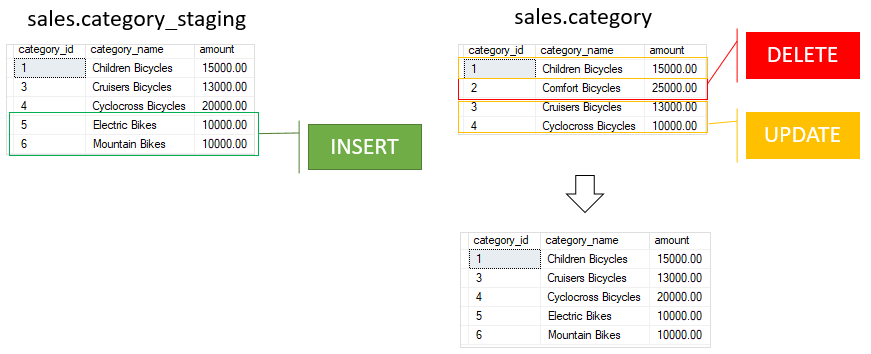


**مثالی از UPDATE وDELETE و INSERT**

|  |
| --- |
| CREATE SCHEMA Sales  go  CREATE TABLE Sales.category (category\_id INT PRIMARY KEY,category\_name VARCHAR(255) NOT NULL,amount DECIMAL(10, 2));  INSERT INTO Sales.category(category\_id, category\_name, amount)  VALUES(1, 'Children Bicycles', 15000),  (2, 'Comfort Bicycles', 25000),  (3, 'Cruisers Bicycles', 13000),  (4, 'Cyclocross Bicycles', 10000);  CREATE TABLE Sales.category\_staging (category\_id INT PRIMARY KEY,category\_name VARCHAR(255) NOT NULL,amount DECIMAL(10, 2));  INSERT INTO Sales.category\_staging(category\_id, category\_name, amount)  VALUES(1, 'Children Bicycles', 15000),  (3, 'Cruisers Bicycles', 13000),  (4, 'Cyclocross Bicycles', 20000),  (5, 'Electric Bikes', 10000),  (6, 'Mountain Bikes', 10000); |

|  |
| --- |
| SELECT \* FROM sales.category  SELECT \* FROM sales.category\_staging |

|  |
| --- |
| MERGE Sales.category t  USING(SELECT b.category\_id, b.category\_name, b.amount FROM Sales.category\_staging b) s  ON(s.category\_id=t.category\_id)  WHEN MATCHED THEN  UPDATE SET t.category\_name=s.category\_name, t.amount=s.amount  WHEN NOT MATCHED BY TARGET THEN  INSERT(category\_id, category\_name, amount)VALUES(s.category\_id, s.category\_name, s.amount)  WHEN NOT MATCHED BY SOURCE THEN DELETE; |



**مثالی از INSERT و UPDATE**

|  |
| --- |
| MERGE Sales.category AS target  USING  (  SELECT \* FROM Sales.category\_staging  ) AS source  ON (target.category\_id = source.category\_id)  WHEN MATCHED THEN  UPDATE SET    category\_name = source.category\_name,  amount = source.amount  WHEN NOT MATCHED THEN  INSERT (category\_id , category\_name , amount )  VALUES (source.category\_id,source.category\_name,source.amount) ; |

**مثالی از DELETE**

|  |
| --- |
| MERGE Sales.category AS target  USING  (  SELECT \* FROM Sales.category\_staging  ) AS source  ON (target.category\_id = source.category\_id)  WHEN NOT MATCHED BY SOURCE THEN  DELETE; |