



5.5:Data Insertion, Updating and Deletion

IT2306 – Database Systems I

Level I - Semester 2

Detailed Syllabus

- 5.5.1 Inserting Data:
 - INSERT INTO [VALUES|SELECT] including a column list, null values;
 - obtaining values from a SELECT.
- 5.5.2 Updating Data:
 - UPDATE (selected columns, selected rows, with a sub query).
- 5.5.3 Deleting Data:
 - DELETE (all data, selected data, with a sub query)

Command: INSERT

Function

- *Places data one or more rows into a table*
- *Data can also be downloaded from another computer system or collected from other sites.*

```
INSERT INTO table-name (column-name),  
| VALUES ([constant, NULL],)
```

or

```
| SELECT retrieval condition
```

Command: INSERT

- Single-Row Insert

```
INSERT INTO Employee (Emp_No, Emp_Name, Age,  
                        Dept)  
VALUES ('E1', 'Dias', 26, 'PER')
```

- Multi-Row Insert

```
INSERT INTO Manager (Emp_No, Emp_Name, Age, Dept)  
SELECT Emp_No, Emp_Name, Age, Dept  
FROM Employee  
WHERE Job = 'Manager'
```

RESTRICT INSERT

Insert with referential integrity

In Employee Table:

```
CONSTRAINT Emp_Dep_FK  
FOREIGN KEY (Dept) REFERENCES  
Department(Dept_Code)
```

```
INSERT INTO Employee  
VALUES (342, 'Dias', 26, 'Sale');
```

An employee can only be inserted if its department is found in department table

RESTRICT INSERT

Department

Dept_Code	Dep_Name	Manager
SAL	Sales	179
FIN	Finance	857



Employee

Emp_N	Emp_Nam	e Age	Dept
179	Silva	27	SAL
857	Perera	34	FIN
342	Dias	26	Sale



Command: UPDATE

Function

- *Changes data in one or more rows of a table*

UPDATE *table-name*
SET (*column-name = expression*),
WHERE *search-condition*

Example

```
UPDATE STUDCLASS  
SET FEES = 1200  
WHERE STUDNO = 1234
```

Selective Update

```
UPDATE STUDCLASS  
SET FEES = 1200
```

Update All Rows

Command: UPDATE

Example

Update with Subquery

```
UPDATE Works_On  
SET Hours = 12  
WHERE Proj_No IN(SELECT Proj_No FROM Project  
WHERE Proj_Name = 'INFORMATION TECHNOLOGY')
```

```
UPDATE    Employee  
SET       Age = Age+1
```


RESTRICT UPDATE

Update with referential integrity

In Employee Table

```
CONSTRAINT Emp_Dep_FK  
FOREIGN KEY (Dept) REFERENCES  
  Department(Dept_Code) ON UPDATE RESTRICT
```

```
UPDATE Department SET Dept_Code = 'Sale'  
WHERE Dept_Code = 'SAL'
```

A department code can only be changed if it is not found in employee table (i.e. no employees working for them)

RESTRICT UPDATE

Department

Dept_Code	Dep_Name	Manager
SAL	Sales	179
FIN	Finance	857



Employee

Emp_No	Emp_Name	Age	Dept
179	Silva	27	SAL
857	Perera	34	FIN
342	Dias	26	SAL



CASCADE UPDATE

Update with referential integrity

In Employee Table

```
CONSTRAINT Emp_Dep_FK  
FOREIGN KEY (Dept) REFERENCES  
Department(Dept_Code) ON UPDATE CASCADE
```

```
UPDATE Department SET Dept_Code = 'Sale'  
WHERE Dept_Code = 'SAL'
```

Updating a department code will result in changing it in the employee table (update with new department code for the employees working for them)

CASCADE UPDATE

Department

Dept_Code	Dep_Name	Manager
Sale	Sales	179
FIN	Finance	857



Employee

Emp_No	Emp_Name	Age	Dept
179	Silva	27	Sale
857	Perera	34	FIN
342	Dias	26	Sale



SET NULL UPDATE

Update with referential integrity

In Employee Table

```
CONSTRAINT Emp_Dep_FK  
FOREIGN KEY (Dept) REFERENCES  
  Department(Dept_Code) ON UPDATE SET NULL
```

```
UPDATE Department SET Dept_Code = 'Sale'  
WHERE Dept_Code = 'SAL'
```

Updating a department code will result in changing the department code of their employees to NULL (only if NULL values are allowed)

SET NULL UPDATE

Department

Dept_Code	Dep_Name	Manager
Sale	Sales	179
FIN	Finance	857



Employee

Emp_No	Emp_Name	Age	Dept
179	Silva	27	NULL
857	Perera	34	FIN
342	Dias	26	NULL



SET DEFAULT UPDATE

Update with referential integrity

In Employee Table

```
CONSTRAINT Emp_Dep_FK  
FOREIGN KEY (Dept) REFERENCES  
  Department(Dept_Code) ON UPDATE  
                           SET DEFAULT 'XXX'
```

```
UPDATE Department SET Dept_Code = 'Sale'  
WHERE Dept_Code = 'SAL'
```

Updating a department code will result in changing the department code of their employees to a default value

SET DEFAULT UPDATE

Department

Dept_Code	Dep_Name	Manager
Sale	Sales	179
FIN	Finance	857



Employee

E m p _ No	E m p _ N a m e	A g e	D e p t
1 7 9	S i l v a	2 7	X X X
8 5 7	P e r e r a	3 4	F I N
3 4 2	D i a s	2 6	X X X

Command: DELETE

Function

- *Removes one or more rows from a table*

DELETE FROM *table-name*
{WHERE *search-condition*}

Example

DELETE FROM Employee
WHERE Emp_No = 'E1'

Select Delete

DELETE FROM Employee

Delete All Rows

DELETE FROM Dependent
WHERE Emp_No = (SELECT Emp_No FROM Employee
WHERE Emp_Name = 'Dias')

Delete with Subquery

RESTRICT DELETE

Delete with referential integrity

In Employee Table

```
CONSTRAINT Emp_Dep_FK  
FOREIGN KEY (Dept) REFERENCES  
Department(Dept_Code) ON DELETE RESTRICT
```

```
DELETE FROM Department  
WHERE Dept_Code = 'SAL'
```

A department can only be deleted if it is not found in employee table (i.e. no employees working for them)

RESTRICT DELETE

Department

Dept_Code	Dep_Name	Manager
SAL	Sales	179
FIN	Finance	857



Employee

Emp_No	Emp_Name	Age	Dept
179	Silva	27	SAL
857	Perera	34	FIN
342	Dias	26	SAL



CASCADE DELETE

Delete with referential integrity

In Employee Table

```
CONSTRAINT Emp_Dep_FK  
FOREIGN KEY (Dept) REFERENCES  
Department(Dept_Code) ON DELETE CASCADE
```

```
DELETE FROM Department  
WHERE Dept_Code = 'SAL'
```

Deleting a department will result in deleting it from the employee table (delete employees working for them)

CASCADE DELETE

Department

Dept_Cod	e Dep_Name	Manager
SAL	Sales	179
FIN	Finance	857



Employee

Emp_No	Emp_Name	Age	Dept
179	Silva	27	SAL
857	Perera	34	FIN
342	Dias	26	SAL



SET NULL DELETE

Delete with referential integrity

In Employee Table

```
CONSTRAINT Emp_Dep_FK  
FOREIGN KEY (Dept) REFERENCES  
Department(Dept_Code) ON DELETE SET NULL
```

```
DELETE FROM Department  
WHERE Dept_Code = 'SAL'
```

Deleting a department will result in changing the department of their employees in the employee table to **NULL** (only if **NULL** values are allowed)

SET NULL DELETE

Department

Dept_Code	Dep_Name	Manager
SAL	Sales	179
FIN	Finance	857



Employee

Emp_No	Emp_Name	Age	Dept
179	Silva	27	NULL
857	Perera	34	FIN
342	Dias	26	NULL

SET DEFAULT DELETE

Delete with referential integrity

In Employee Table

```
CONSTRAINT Emp_Dep_FK  
FOREIGN KEY (Dept) REFERENCES  
  Department(Dept_Code) ON DELETE  
                           SET DEFAULT 'XXX'
```

```
DELETE FROM Department  
WHERE Dept_Code = 'SAL'
```

Deleting a department will result in changing the department of their employees in the employee table to a specified default value


SET DEFAULT DELETE

Department

Dept_Code	Dept_Name	Manager
SAL	Sales	179
FIN	Finance	857



Employee



Emp_No	Emp_Name	Age	Dept
179	Silva	27	XXX
857	Perera	34	FIN
342	Dias	26	XXX