Week 13

Database Systems -

Introduction to Databases and Data Warehouses

CHAPTER 6 - Database Implementation and Use (Part 1)

MAIN TOPIC

- Implement Referential Integrity Constraints

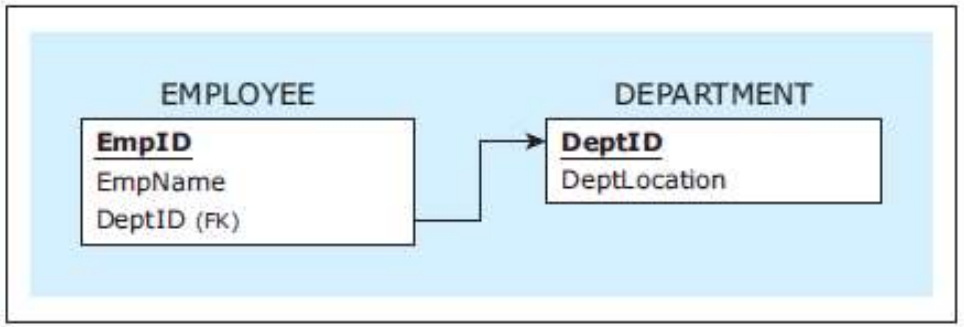
REFERENTIAL INTEGRITY CONSTRAINT

- Referential integrity constraint
  - *In each row of a relation containing a foreign key, the value of the **foreign key EITHER matches** one of the values in the **primary key** column of the referred relation **OR** the value of **the foreign key is null** (empty).*
  - Regulates the relationship between a table with a foreign key and a table with a primary key to which the foreign key refers

REFERENTIAL INTEGRITY CONSTRAINT

Referential integrity constraint compliance example

Two relations and a referential integrity constraint



Data records in compliance with the referential integrity constraint

EMPLOYEE

EmpID	EmpName	DeptID
1234	Becky	1
2345	Molly	2
3456	Rob	1
1324	Ted	2

DEPARTMENT

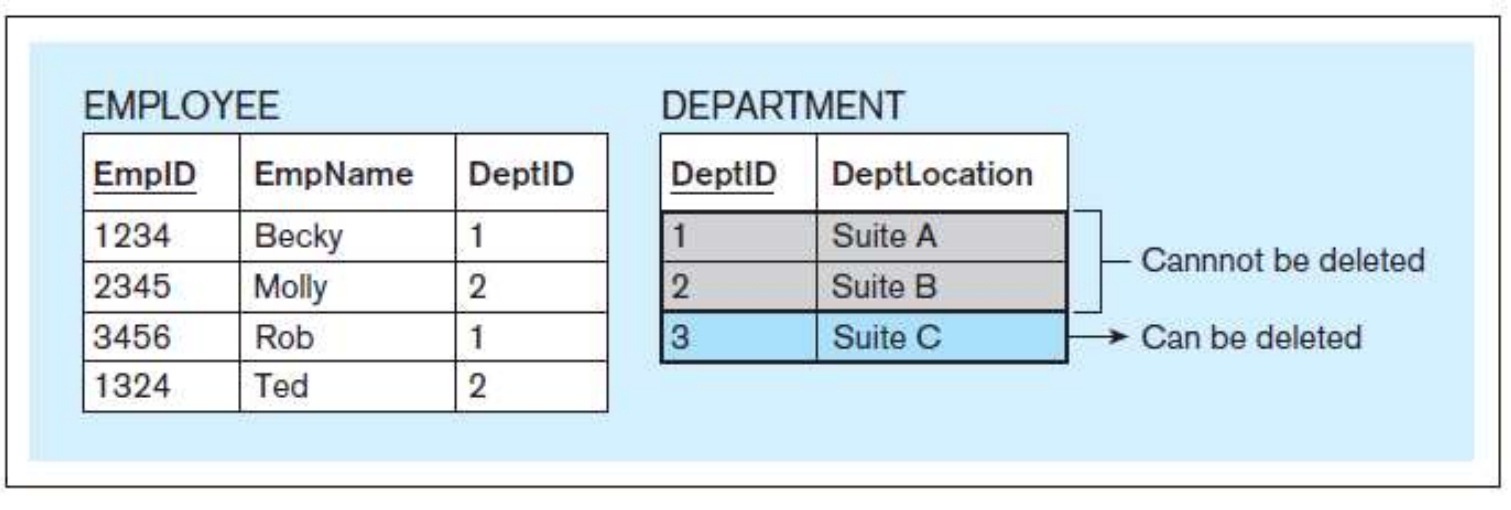
DeptID	DeptLocation
1	Suite A
2	Suite B
3	Suite C

REFERENTIAL INTEGRITY CONSTRAINT: DELETE AND UPDATE IMPLEMENTATION OPTIONS

- Referential integrity constraint - delete and update implementation options
  - Delete options
    - DELETE RESTRICT
    - DELETE CASCADE
    - DELETE SET-TO-NULL
    - DELETE SET-TO-DEFAULT
  - Update options
    - UPDATE RESTRICT
    - UPDATE CASCADE
    - UPDATE SET-TO-NULL
    - UPDATE SET-TO-DEFAULT

REFERENTIAL INTEGRITY CONSTRAINT: DELETE AND UPDATE IMPLEMENTATION OPTIONS

DELETE RESTRICT example

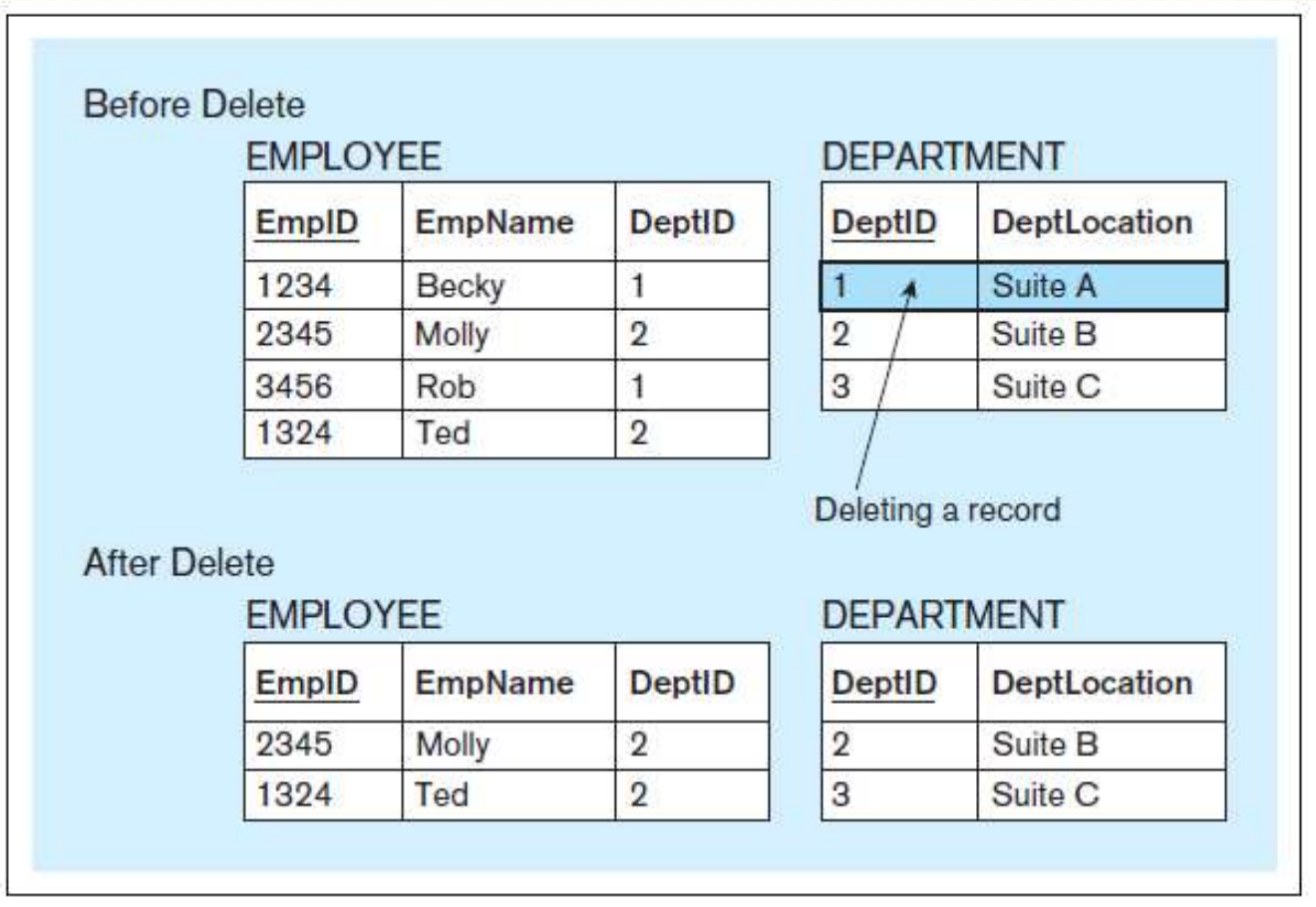


Can not delete primary key values if they are referred by foreign key values.

REFERENTIAL INTEGRITY CONSTRAINT: DELETE AND UPDATE IMPLEMENTATION OPTIONS

DELETE CASCADE example

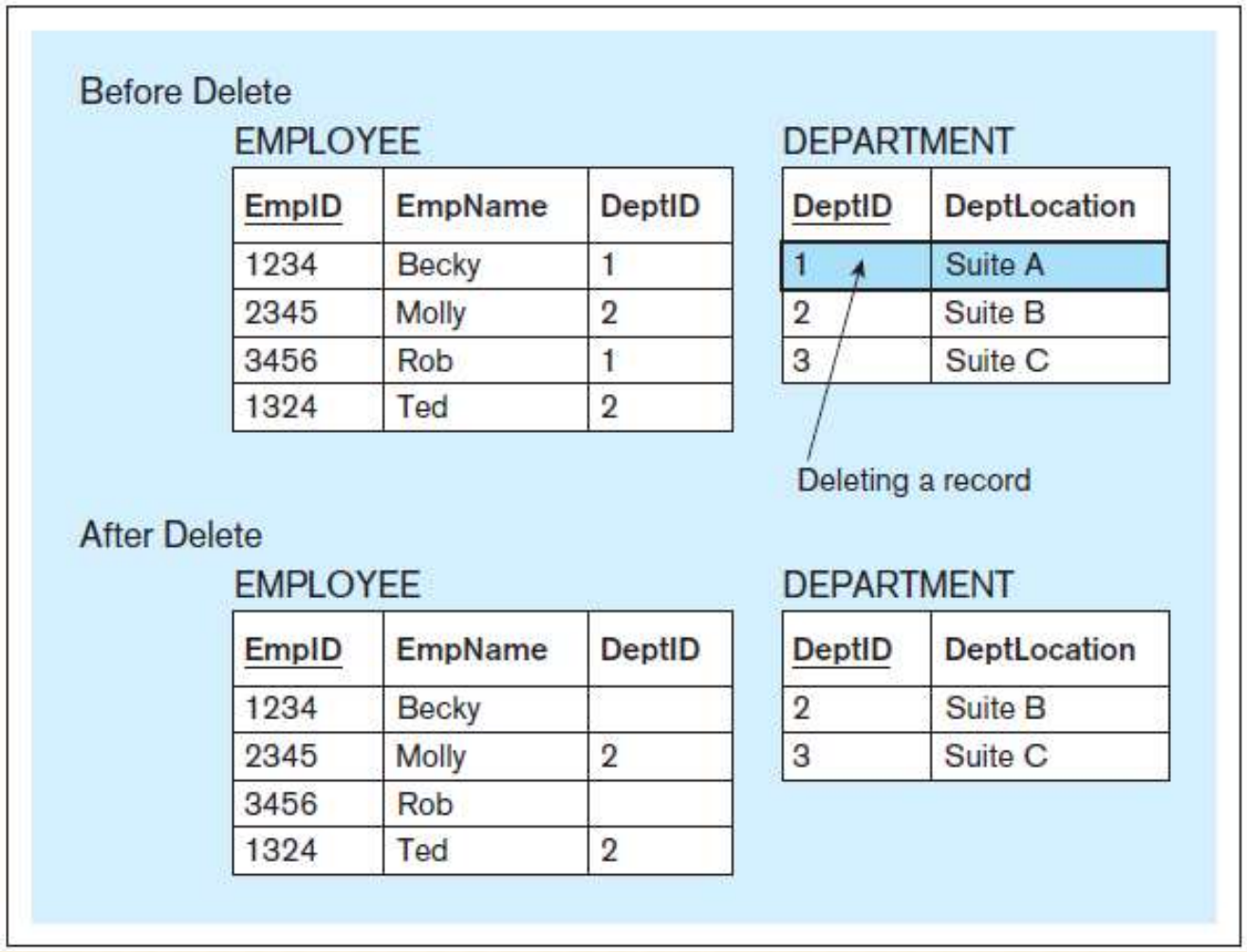
Deleting primary key values also deletes all related foreign key values.



REFERENTIAL INTEGRITY CONSTRAINT: DELETE AND UPDATE IMPLEMENTATION OPTIONS

DELETE SET-TO-NULL example

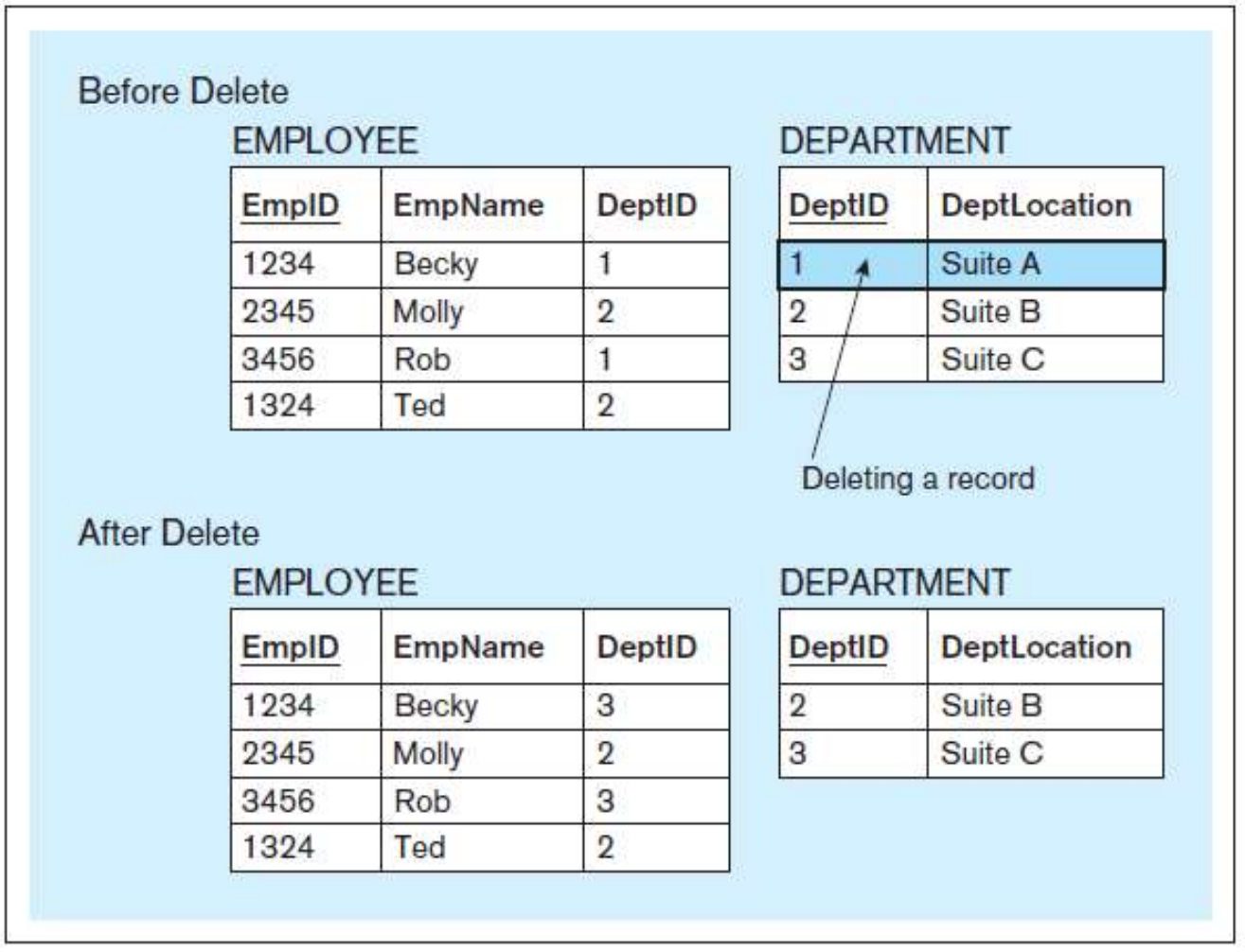
Deleting primary key values sets all related foreign key values to NULL.



REFERENTIAL INTEGRITY CONSTRAINT: DELETE AND UPDATE IMPLEMENTATION OPTIONS

DELETE SET-TO-DEFAULT example

Deleting primary key values sets all related foreign key values to a default value.

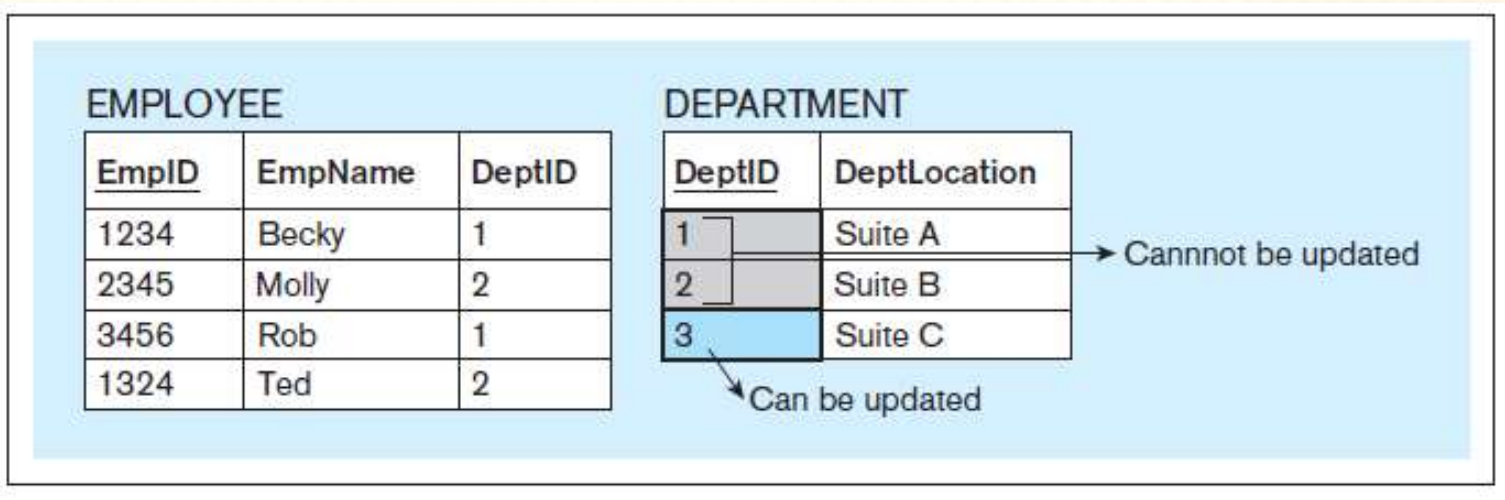


REFERENTIAL INTEGRITY CONSTRAINT: DELETE AND UPDATE IMPLEMENTATION OPTIONS

- Referential integrity constraint
  - Delete implementation options only affect the data deletion in the referred relation that contains a primary key referenced by foreign keys.
  - Delete implementation options does NOT affect the data deletion in the referring relation that contains the foreign keys.
    - E.g. deleting employees in relation EMPLOYEE is not affected by delete implementation options

REFERENTIAL INTEGRITY CONSTRAINT: DELETE AND UPDATE IMPLEMENTATION OPTIONS

UPDATE RESTRICT example

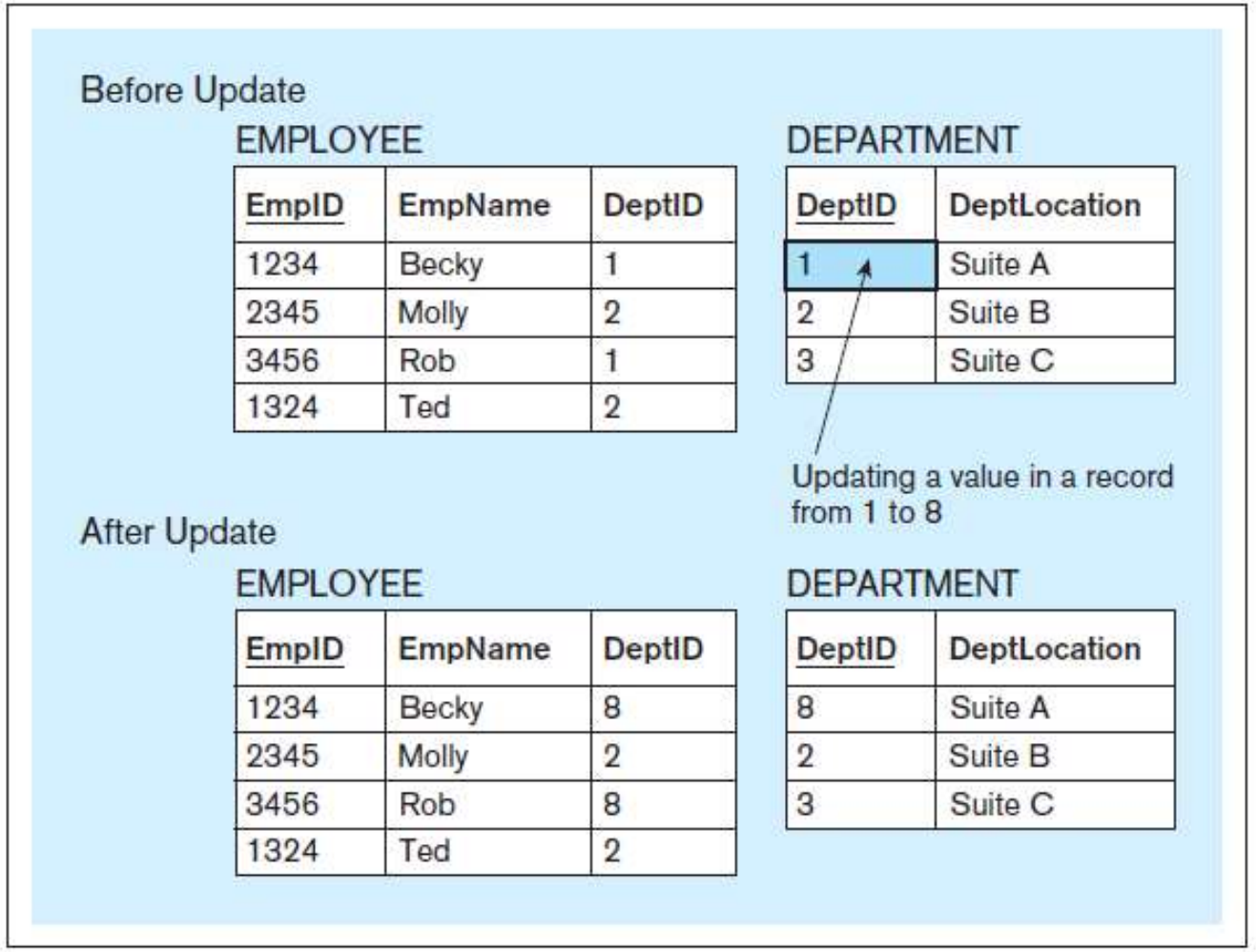


Can not modify primary key values if they are referred by foreign key values.

REFERENTIAL INTEGRITY CONSTRAINT: DELETE AND UPDATE IMPLEMENTATION OPTIONS

UPDATE CASCADE example

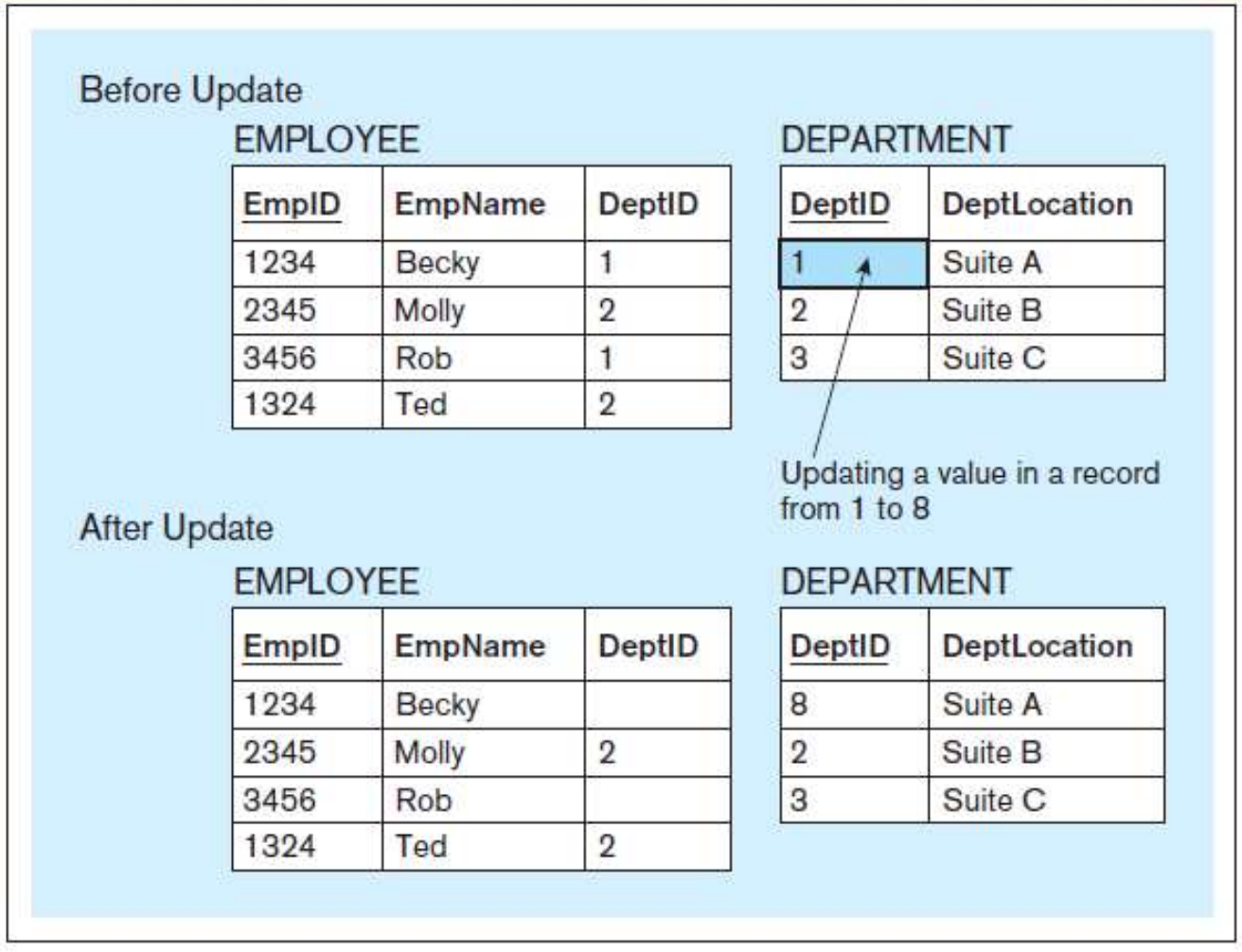
Updating primary key values also updates all related foreign key values to the same values.



REFERENTIAL INTEGRITY CONSTRAINT: DELETE AND UPDATE IMPLEMENTATION OPTIONS

UPDATE SET-TO-NULL example

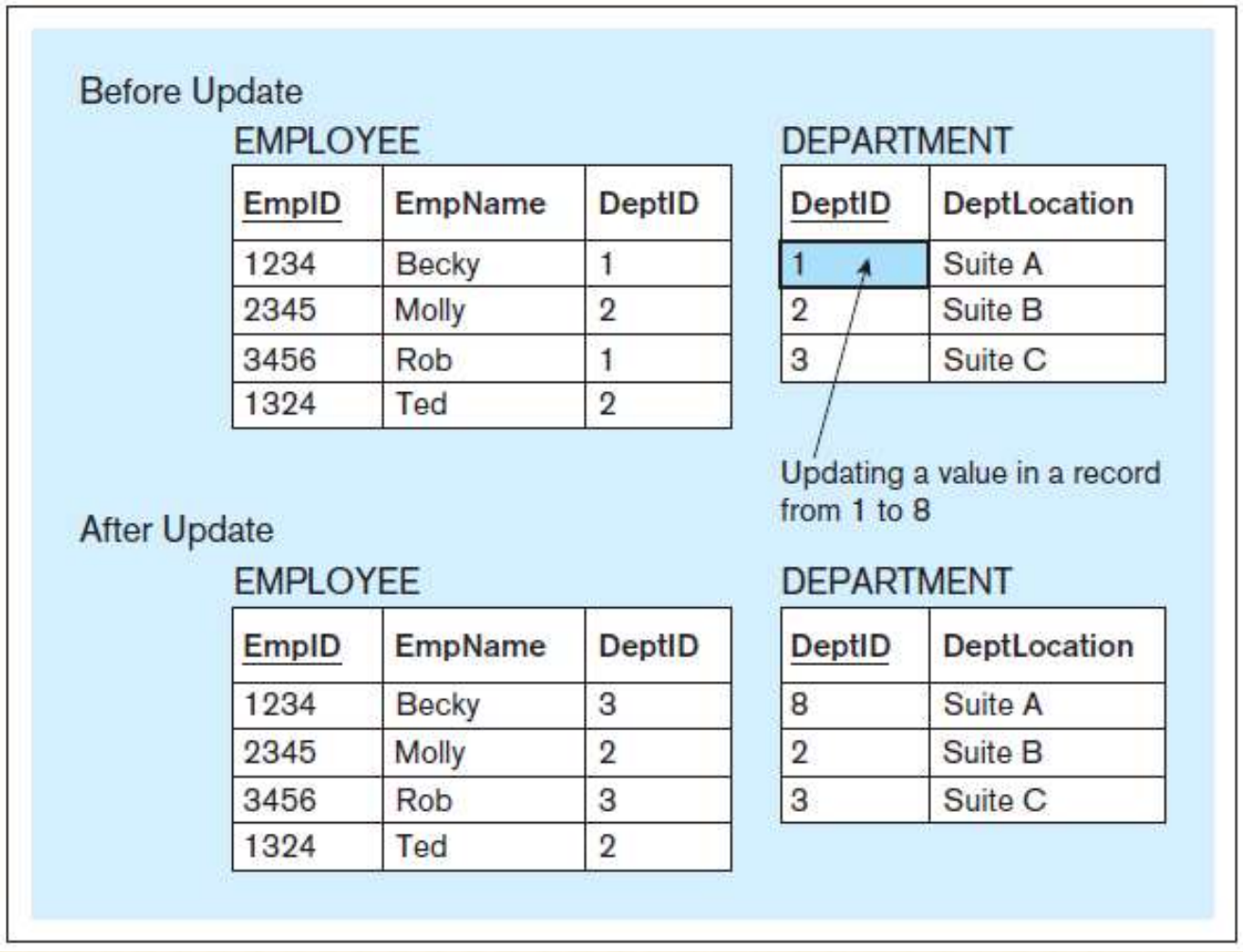
Updating primary key values sets all related foreign key values to NULL.



REFERENTIAL INTEGRITY CONSTRAINT: DELETE AND UPDATE IMPLEMENTATION OPTIONS

UPDATE SET-TO-DEFAULT example

Updating primary key values sets all related foreign key values to a default value.



REFERENTIAL INTEGRITY CONSTRAINT: DELETE AND UPDATE IMPLEMENTATION OPTIONS

- Implementing delete and update options
  - Example - DELETE RESTRICT and UPDATE RESTRICT

```
CREATE TABLE employee
( empid CHAR(4),
empname CHAR(20),
deptid CHAR(2),
PRIMARY KEY (empid),
FOREIGN KEY (deptid) REFERENCES department);
```

  - By default, the options are DELETE RESTRICT and UPDATE RESTRICT
  - Chapter examples are for the illustration purpose.
  - Syntax in different RDBMS products may vary.

REFERENTIAL INTEGRITY CONSTRAINT: DELETE AND UPDATE IMPLEMENTATION OPTIONS

- Implementing delete and update options
  - Example - DELETE RESTRICT and UPDATE RESTRICT
  - By default, the options are DELETE RESTRICT and UPDATE RESTRICT
    - Example: Chapter 5 ZAGI database

```
CREATE TABLE product
( productid CHAR(3) NOT NULL,
productname VARCHAR(25) NOT NULL,
productprice NUMERIC (7,2) NOT NULL,
vendorid CHAR(2) NOT NULL,
categoryid CHAR(2) NOT NULL,
PRIMARY KEY (productid),
FOREIGN KEY (vendorid) REFERENCES vendor(vendorid),
FOREIGN KEY (categoryid) REFERENCES
category(categoryid) );
```

    - Try deleting vendor with id ‘PG’

REFERENTIAL INTEGRITY CONSTRAINT: DELETE AND UPDATE IMPLEMENTATION OPTIONS

- Implementing delete and update options
  - Example - DELETE CASCADE

```
CREATE TABLE employee
( empid CHAR(4),
empname CHAR(20),
deptid CHAR(2),
PRIMARY KEY (empid)
FOREIGN KEY (deptid) REFERENCES department
                     ON DELETE CASCADE);
```

REFERENTIAL INTEGRITY CONSTRAINT: DELETE AND UPDATE IMPLEMENTATION OPTIONS

- Implementing delete and update options
  - Example – UPDATE SET-TO-NULL

```
CREATE TABLE employee
( empid CHAR(4),
empname CHAR(20),
deptid CHAR(2),
PRIMARY KEY (empid)
FOREIGN KEY (deptid) REFERENCES department
                     ON UPDATE SET NULL);
```

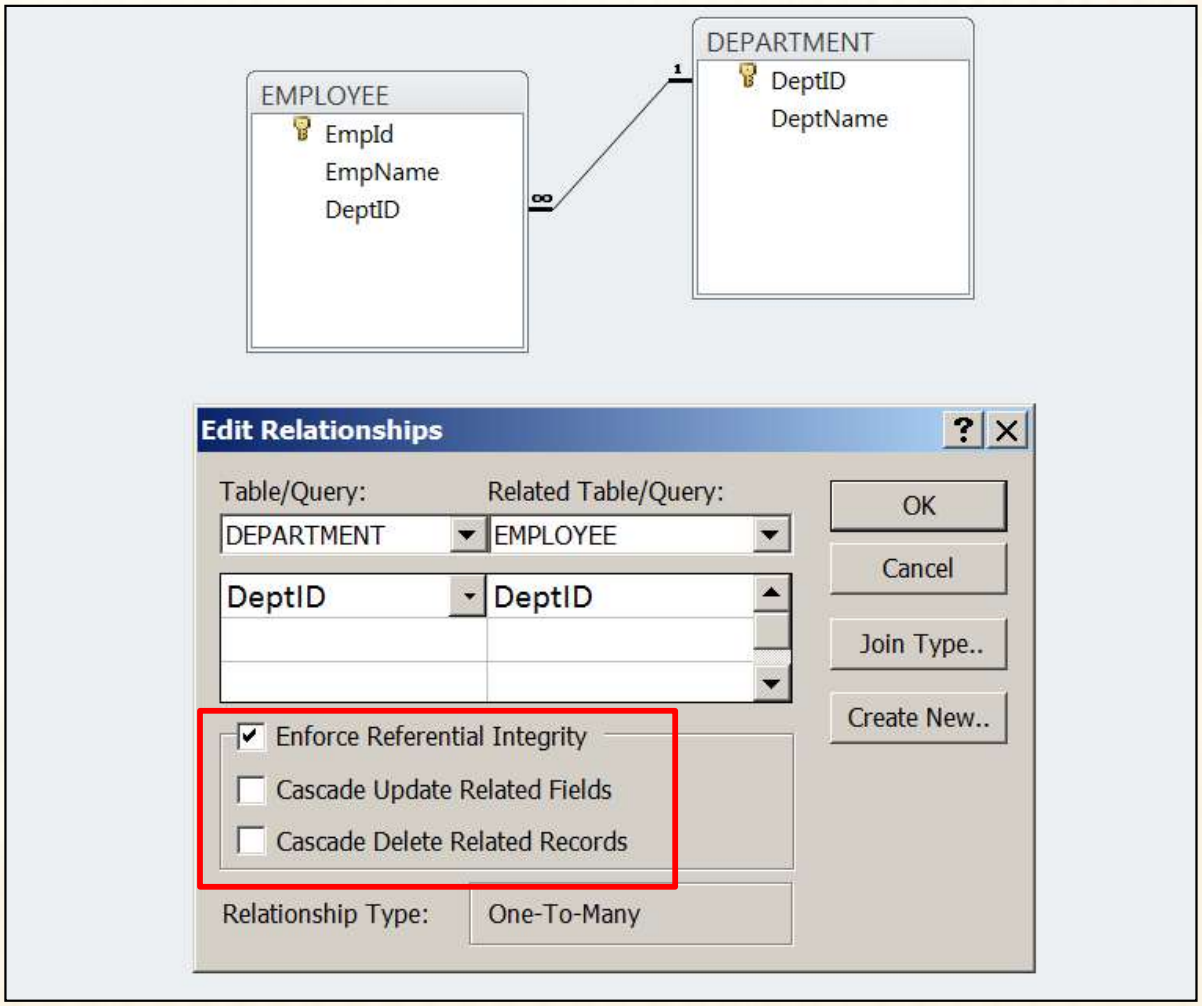
REFERENTIAL INTEGRITY CONSTRAINT: DELETE AND UPDATE IMPLEMENTATION OPTIONS

- Implementing delete and update options
  - Example - DELETE CASCADE and UPDATE SET-TO-NULL

```
CREATE TABLE employee
( empid CHAR(4),
empname CHAR(20),
deptid CHAR(2),
PRIMARY KEY (empid)
FOREIGN KEY (deptid) REFERENCES department
                     ON DELETE CASCADE
                     ON UPDATE SET NULL);
```

REFERENTIAL INTEGRITY CONSTRAINT: DELETE AND UPDATE IMPLEMENTATION OPTIONS

- Implementing delete and update options
  - Example - DELETE RESTRICT and UPDATE RESTRICT (MS Access)



IMPLEMENTING USER-DEFINED CONSTRAINTS

- Implementing user-defined constraints
  - Methods for implementing user-defined constraints include:
    - CHECK clause
    - Assertions and triggers
    - Coding in specialized database programming languages that combine SQL with additional non-SQL statements for processing data from databases (such as PL/SQL)
    - Embedding SQL with code written in regular programming languages (such as C++ or Java)