

Database Management Systems

SQL Query Language (2)

Topics

- Update Query
- Delete Query
 - Integrity Constraint
 - Cascade Deletes
 - Deleting a Table
- Join in Queries
- More Options in Select Queries
 - Like Clause
 - Distinct Queries
 - Ordering

Update Query

- Update query is used for changing the values of attributes.

- *Update* Table_name

Set Attribute_name1 = Value1,
Attribute_name2 = Value2,

.....

Where condition

Update Query Example

- Raise the salary of every employee by 20% if the salary is less than or equal to 40

update Employee

set Salary = Salary * 1.2

where Salary <= 40

Update Example

EMPLOYEE	FirstName	Surname	Dept	Office	Salary	City
	Mary	Brown	Administration	10	45	London
	Charles	White	Production	20	36	Toulouse
	Gus	Green	Administration	20	40	Oxford
	Jackson	Neri	Distribution	16	45	Dover
	Charles	Brown	Planning	14	80	London
	Laurence	Chen	Planning	7	73	Worthing
	Pauline	Bradshaw	Administration	75	40	Brighton
	Alice	Jackson	Production	20	46	Toulouse

EMPLOYEE	FirstName	Surname	Dept	Office	Salary	City
	Mary	Brown	Administration	10	45	London
	Charles	White	Production	20	36 43.2	Toulouse
	Gus	Green	Administration	20	40 48	Oxford
	Jackson	Neri	Distribution	16	45	Dover
	Charles	Brown	Planning	14	80	London
	Laurence	Chen	Planning	7	73	Worthing
	Pauline	Bradshaw	Administration	75	40 48	Brighton
	Alice	Jackson	Production	20	46	Toulouse

Update Query Order

- Update Query is set oriented. Therefore the order of the queries is important.
- For example the following order gives some employees a double raise

```
update Employee  
set Salary = Salary * 1.2  
where Salary <= 40
```

```
update Employee  
set Salary = Salary * 1.15  
where Salary > 40
```

Delete Query

- To delete tuples *Delete* query is used.
- *Delete from* Table_name
 Where condition
- *Delete* query without *Where* deletes all tuples. (Table schema is preserved)

Delete Query Example

- Delete From Employee
Where City = 'London'

EMPLOYEE	FirstName	Surname	Dept	Office	Salary	City
	Mary	Brown	Administration	10	45	London
	Charles	White	Production	20	36	Toulouse
	Gus	Green	Administration	20	40	Oxford
	Jackson	Neri	Distribution	16	45	Dover
	Charles	Brown	Planning	14	80	London
	Laurence	Chen	Planning	7	73	Worthing
	Pauline	Bradshaw	Administration	75	40	Brighton
	Alice	Jackson	Production	20	46	Toulouse

Integrity Constraint

- If a table contains a foreign key attribute, the value of the foreign key must refer to a valid record in the second table. This constraint is called *Integrity Constraint*

Sample Database

EMPLOYEE

FirstName	Surname	DeptCode	Office	Salary	City
Mary	Brown	1001	10	45	London
Charles	White	1002	20	36	Toulouse
Gus	Green	1001	20	40	Oxford
Jackson	Neri	1003	16	45	Dover
Charles	Brown	1004	14	80	London
Laurence	Chen	1004	7	73	Worthing
Pauline	Bradshaw	1001	75	40	Brighton
Alice	Jackson	1002	20	46	Toulouse

DEPARTMENT

DeptCode	DeptName	Address	City
1001	Administration	Bond Street	London
1002	Production	Rue Victor Hugo	Toulouse
1003	Distribution	Pond Road	Brighton
1004	Planning	Bond Street	London
1005	Research	Sunset Street	San José

Integrity Constraint In Sample Database

- DeptCode in EMPLOYEE table is a foreign key referring to DeptCode in DEPARTMENT table
- If a department is deleted, its employees will not have a valid department

Integrity Constraint Violation Example

Delete From Department

Where DepartmentName = 'Administration'

After this query, Mary Brown's department is invalid. The database integrity is violated after this query.

Solution to Integrity Violation: Cascade Delete

- To preserve database integrity we should either
 - Not allow deleting records with references from other tables
 - Delete records in other tables which are referring to this record. This is called cascade delete
 - Syntax: *Delete From* tableName
where condition *Cascade*

Cascade Delete Example

EMPLOYEE

FirstName	Surname	DeptCode	Office	Salary	City
Mary	Brown	1001	10	45	London
Charles	White	1002	20	36	Toulouse
Gus	Green	1001	20	40	Oxford
Jackson	Neri	1003	16	45	Dover
Charles	Brown	1004	14	80	London
Laurence	Chen	1004	7	73	Worthing
Pauline	Bradshaw	1001	75	40	Brighton
Alice	Jackson	1002	20	46	Toulouse

DEPARTMENT

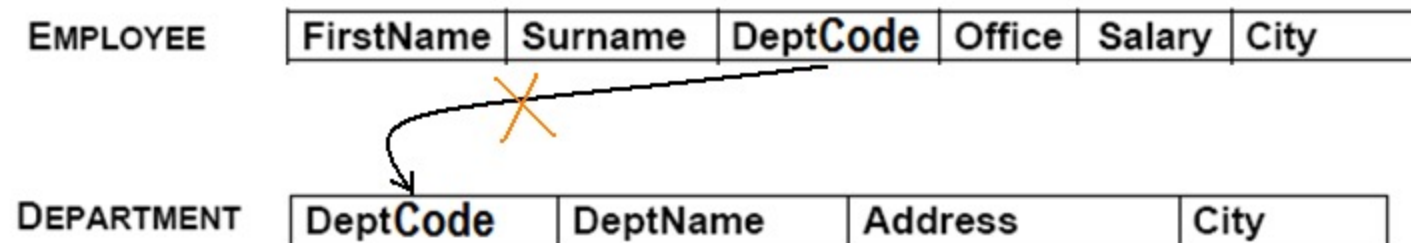
DeptCode	DeptName	Address	City
1001	Administration	Bond Street	London
1002	Production	Rue Victor Hugo	Toulouse
1003	Distribution	Pond Road	Brighton
1004	Planning	Bond Street	London
1005	Research	Sunset Street	San José

Deleting a Table

- A table including its schema and records can be deleted using ***Drop*** command
- Syntax: ***Drop Table*** TableName
 - e.g. Drop Table Cars
- If Cascade is used with Drop command, all references to the table will be removed

Deleting Tables: Example

- Drop Table Department Cascade



Join in Queries

- Join can be used in SQL queries
- Syntax:

SELECT AttributesList

FROM Table1 [JoinType] ***Join*** Table2 ***On*** JoinCond

Where Condition

Example: Natural Join

EMPLOYEE	FirstName	Surname	Dept	Office	Salary	City
	Mary	Brown	Administration	10	45	London
	Charles	White	Production	20	36	Toulouse
	Gus	Green	Administration	20	40	Oxford
	Jackson	Neri	Distribution	16	45	Dover
	Charles	Brown	Planning	14	80	London
	Laurence	Chen	Planning	7	73	Worthing
	Pauline	Bradshaw	Administration	75	40	Brighton
	Alice	Jackson	Production	20	46	Toulouse

DEPARTMENT	DeptName	Address	City
	Administration	Bond Street	London
	Production	Rue Victor Hugo	Toulouse
	Distribution	Pond Road	Brighton
	Planning	Bond Street	London
	Research	Sunset Street	San José

Join Example

- **Select** FirstName, Surname, Department.City
From Employee **join** Department **on** Dept = DeptName

FirstName	Surname	City
Mary	Brown	London
Charles	White	Toulouse
Gus	Green	London
Jackson	Neri	Brighton
Charles	Brown	London
Laurence	Chen	London
Pauline	Bradshaw	London
Alice	Jackson	Toulouse

Left Join, Right Join, Full Join

- Use Left, Right, or Full as join type to have a left, right, or full join
- e.g.

*Select **

From Table1 *Left Join* Table2 *on* Att1=Att2

Where Table1.Att3 > 100

Left Join Example

Select FirstName, Surname, Driver.DriverID,
CarRegNo, Make, Model
From Driver **left join** Automobile **on**
(Driver.DriverID = Automobile.DriverID)

DRIVER	FirstName	Surname	DriverID
	Mary	Brown	VR 2030020Y
	Charles	White	PZ 1012436B
	Marco	Neri	AP 4544442R

AUTOMOBILE	CarRegNo	Make	Model	DriverID
	ABC 123	BMW	323	VR 2030020Y
	DEF 456	BMW	Z3	VR 2030020Y
	GHI 789	Lancia	Delta	PZ 1012436B
	BBB 421	BMW	316	MI 2020030U

The Result of Left Join Example

FirstName	Surname	DriverID	CarRegNo	Make	Model
Mary	Brown	VR 2030020Y	ABC 123	BMW	323
Mary	Brown	VR 2030020Y	DEF 456	BMW	Z3
Charles	White	PZ 1012436B	GHI 789	Lancia	Delta
Marco	Neri	AP 4544442R	NULL	NULL	NULL

Right Join Example

- **Select** FirstName, Surname, Driver.DriverID,
CarRegNo, Make, Model
From Driver **right join** Automobile **on**
(Driver.DriverID = Automobile.DriverID)

FirstName	Surname	DriverID	CarRegNo	Make	Model
Mary	Brown	VR 2030020Y	ABC 123	BMW	323
Mary	Brown	VR 2030020Y	DEF 456	BMW	Z3
Charles	White	PZ 1012436B	GHI 789	Lancia	Delta
NULL	NULL	MI 2020030U	BBB 421	BMW	316

Full Join Example

- **Select** FirstName, Surname, Driver.DriverID,
CarRegNo, Make, Model
From Driver **full join** Automobile **on**
(Driver.DriverID = Automobile.DriverID)

FirstName	Surname	DriverID	CarRegNo	Make	Model
Mary	Brown	VR 2030020Y	ABC 123	BMW	323
Mary	Brown	VR 2030020Y	DEF 456	BMW	Z3
Charles	White	PZ 1012436B	GHI 789	Lancia	Delta
Marco	Neri	AP 4544442R	NULL	NULL	NULL
NULL	NULL	MI 2020030U	BBB 421	BMW	316

Like Clause

- Like is used in condition expressions when a pattern is used instead of an exact value.
- `_` in like represents one character
- `%` in like represents any number of characters
 - e.g. `A%A` matches all strings starting and ending with character 'A' such as 'ANKARA'
 - e.g. `_an` matches all 3 letter strings ending with 'an' such as 'man' or 'can'

Like Clause Example

- Find the employees with surnames that have 'r' as the second letter and 'n' as the last letter:

```
select *  
from Employee  
where Surname like '_r%n'
```

Distinct Option in Queries

- In SQL, tables may have identical rows if no primary key is defined
- Duplicates can be removed from a query using the keyword *distinct*

Query with Distinct Keyword

```
select City  
from Department
```

City
London
Toulouse
Brighton
London
San José

```
select distinct City  
from Department
```

City
London
Toulouse
Brighton
San José

Ordering

- The *order by* clause, at the end of the query, orders the rows of the result.

- Syntax:

order by Attribute [asc | desc]

- More than one attribute can be used in *order by* clause

Ordering Example

- Extract all records from the Automobile table in descending order of make and model:

Select *

From Automobile

Order by Make desc, Model desc

CarRegNo	Make	Model	DriverID
GHI 789	Lancia	Delta	PZ 1012436B
DEF 456	BMW	Z3	VR 2030020Y
ABC 123	BMW	323	VR 2030020Y
BBB 421	BMW	316	MI 2020030U

Summary

- SQL queries are also used for updating records, deleting records or deleting tables.
- In deleting records or tables, the integrity of the database should be preserved.
- Queries can be written by joining tables
- The result of a query can be put in order of an attribute and the repeated rows can be removed.

Questions?