



DATABASE SYSTEMS

Dr. Noha Nagy

Lecture 5

SQL : DML

SQL

Structured Query Language

2

- **Data Definition Language (DDL)**
 - ▣ Define relational *schemata*
 - ▣ **Create/Alter/Drop** tables and their attributes
- **Data Manipulation Language (DML)**
 - ▣ **Insert/Delete/Update** tuples in tables
 - ▣ Query one or more table

DML: Data Manipulation Language

3

- **DML** is used to retrieve, insert, update, and/or delete instances in a database
 - **INSERT:** is used to insert new instances inside a database
 - **UPDATE:** is used to update existing instances inside a database
 - **DELETE:** is used to delete existing instances inside a database
 - **SELECT:** is used to retrieve data from a database

SQL INSERT STATEMENT

4

- Adds one or more rows to a table
- **Inserting into a table**

Insert into <Table Name>

VALUES (value1, value2, value3);

- **Inserting a record that has some null attributes requires identifying the fields that get data**

Insert into <Table Name> (column1, column2, column3)

Values (value1, value2, value3);

Example

5

INSERT INTO mynewtable
Values (1, 'Ahmed', 'Cairo');

OR

INSERT INTO mynewtable (id, name, city)
Values (1, 'Ahmed', 'Cairo');

Mynewtable

ID	Name	City



ID	Name	City
1	Ahmed	Cairo

Insert Statment

6

Employee

<u>Enum</u>	Ename	phone	Pnum

Insert into Employee values (128, 'Mahmoud', 01113005581, 326);

Insert into Employee (Enum,Ename, Pnum)values (130, 'Eyad' , 327);

Insert into Employee (Ename,Enum, Pnum)values ('Ali',134 , 327);

Employee

<u>Enum</u>	Ename	phone	Pnum
<u>128</u>	Mahmoud	01113005581	326
<u>130</u>	Eyad		327
<u>134</u>	Ali		327

**Can I insert more than one
record in one insert
statement?**

SQL UPDATE

7

- Modifies data in existing rows

Update TableName

SET columnName = Value, columnName = Value

Where <Condition>

Update Statment

8

Product

<u>Pnum</u>	Pname	Price	Quantity
<u>123</u>	Arial	200	20
<u>124</u>	Persil	180	50
<u>127</u>	OXI	100	11
<u>128</u>	Tide	150	32

Update Product Set Price=price*2



Product

<u>Pnum</u>	Pname	Price	Quantity
<u>123</u>	Arial	400	20
<u>124</u>	Persil	360	50
<u>127</u>	OXI	200	11
<u>128</u>	Tide	300	32

Update Statment

9

Product

<u>Pnum</u>	Pname	Price	Quantity
<u>123</u>	Arial	200	20
<u>124</u>	Persil	180	50
<u>127</u>	OXI	100	11
<u>128</u>	Tide	150	32

Update Product **Set** Quantity= Quantity – 1 **Where** Pnum= 123

Product

<u>Pnum</u>	Pname	Price	Quantity
<u>123</u>	Arial	400	19
<u>124</u>	Persil	360	50
<u>127</u>	OXI	200	11
<u>128</u>	Tide	300	32



SQL Update: Example 1

10

- Modify customer name from Ali to Mahmoud

Update customers set name = 'Mahmoud'
where name = 'Ali'

SQL Update: Example 2

11

Add 10% bonus on the salaries of all employees:

Employees (emp_id, emp_name, salary)

Update Employees

Set salary = salary * 1.1

SQL Update: Example 3

12

- Change the salary of employee 13 to be 5000

Update Employees

Set salary = 5000

Where emp_id = 13

SQL DELETE

13

- ❑ Removes rows from a table
- ❑ Delete certain rows
 - ❑ **DELETE FROM** TableName **WHERE** <condition>;
- ❑ Delete all rows
 - ❑ **DELETE FROM** TableName ;

Delete Statment

14

Employee

<u>Enum</u>	Ename	phone	Pnum
<u>123</u>	Ahmed	01110025878	111
<u>124</u>	Ali	01225929785	254
<u>127</u>	Ola	0102457896	111

Delete From Employee
Where Pnum = 254;

Employee

<u>Enum</u>	Ename	phone	Pnum
<u>123</u>	Ahmed	01110025878	111
<u>127</u>	Ola	0102457896	111

Example

15

Employee

<u>Enum</u>	Ename	phone	Pnum
<u>123</u>	Ahmed	01110025878	111
<u>124</u>	Ali	01225929785	254
<u>127</u>	Ola	0102457896	111

Delete From Employee;

Employee

<u>Enum</u>	Ename	phone	Pnum

Example

16

- Delete all customers from the customers table that are living in Paris

Customers (Id, name, city)

Delete from customers where city='Paris'

Question

17

- Delete all the data from your table “MyCustomers”

- **Delete from MyCustomers;**

Don't do that on real data!!!

- Delete the table itself

Drop table MyCustomers;

The SELECT Statement

18

- Used for queries on **single** or **multiple** tables
- Clauses of the SELECT statement:
 - ▣ **SELECT**
 - List the columns (and expressions) that should be returned from the query
 - ▣ **FROM**
 - Indicate the table(s) or view(s) from which data will be obtained
 - ▣ **WHERE**
 - Indicate the conditions under which a row will be included in the result
 - ▣ **GROUP BY**
 - Indicate categorization of results
 - ▣ **HAVING**
 - Indicate the conditions under which a category (group) will be included
 - ▣ **ORDER BY**
 - Sorts the result according to specified criteria

DML - SQL SELECT Statement

19

- The **SELECT** statement allows you to read data from one or more tables. To write a SELECT statement in MySQL, you follow this syntax:

SELECT select_list

FROM table_name

WHERE conditions

SQL SELECT – Single Column

20

- Using the SELECT statement to retrieve data from a single column example:

- ▣ **SELECT** "column" **FROM** "tablename";

- ▣ **SELECT** lastname **From** employees;

- Using the SELECT statement to query data from multiple columns example:

- ▣ **SELECT** lastname, firstname, jobtitle **FROM** employees;

Retrieve Specific Columns

21

Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

```
SELECT PName, Price
FROM   Product
```



“projection”

PName	Price
Gizmo	\$19.99
Powergizmo	\$29.99
SingleTouch	\$149.99
MultiTouch	\$203.99

SQL SELECT - DISTINCT Keyword

22

- Distinct allow you to remove all the duplicates from the result.
- Select lastName from employees;
- Select **distinct** lastName from employees;

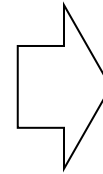
DISTINCT: Eliminating Duplicates

23

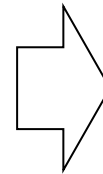
```
SELECT DISTINCT Category  
FROM Product
```

Versus

```
SELECT Category  
FROM Product
```



Category
Gadgets
Photography
Household



Category
Gadgets
Gadgets
Photography
Household

SQL SELECT – All Attributes

24

- Using the MySQL SELECT statement to retrieve data from all columns example:

```
SELECT * FROM employees
```

- Often called “select star” or “select *”

Retrieve All Columns and All Rows

25

Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

SELECT Pname, Price, Category, Manufacturer
FROM Product

OR

SELECT *
FROM Product



PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

SQL WHERE Clause - Operators

26

Operator	Description
=	Equal to. You can use it with almost any data types.
<> or !=	Not equal to
<	Less than. You typically use it with numeric and date/time data types.
>	Greater than.
<=	Less than or equal to
>=	Greater than or equal to

Retrieve Specific Rows

27

Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

```
SELECT *  
FROM   Product  
WHERE  category='Gadgets'
```



“selection”

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks

Retrieve Specific Columns and Rows

28

Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

```
SELECT PName, Price, Manufacturer
FROM   Product
WHERE  Price > 100
```



“selection” and
“projection”

PName	Price	Manufacturer
SingleTouch	\$149.99	Canon
MultiTouch	\$203.99	Hitachi

Multiple Conditions with AND

29

Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	GizmoWorks
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

```
SELECT PName, Price, Manufacturer
FROM   Product
WHERE  Price > 100 and Manufacturer= 'Canon'
```



PName	Price	Manufacturer
SingleTouch	\$149.99	Canon

Multiple Conditions with OR

30

Product

PName	Price	Category	Manufacturer
Gizmo	\$19.99	Gadgets	Samsung
Powergizmo	\$29.99	Gadgets	GizmoWorks
SingleTouch	\$149.99	Photography	Canon
MultiTouch	\$203.99	Household	Hitachi

```
SELECT PName, Price, Manufacturer
FROM   Product
WHERE  Price > 100 OR Manufacturer= 'Samsung'
```



PName	Price	Manufacturer
SingleTouch	\$149.99	Canon
MultiTouch	\$203.99	Hitachi
Gizmo	\$19.99	Samsung

LIKE: Simple String Pattern Matching

31

```
SELECT *  
FROM Products  
WHERE PName LIKE '%Touch%'
```

- ❑ **s LIKE p**: pattern matching on **strings**
- ❑ **S** is a column name
- ❑ **p** may contain two special symbols:
 - ▣ **%** = zero, one, or multiple characters
 - ▣ **_** = any single character

LIKE Cont,

32

Employee

<u>Enum</u>	Ename	phone
<u>123</u>	Ahmed	01110025878
<u>124</u>	Ali	01225929785
<u>127</u>	Ola	0102457896

- selects all Employees with a Name that start with "A"

```
SELECT *  
FROM Employee  
WHERE Ename LIKE 'A%';
```

<u>Enum</u>	Ename	phone
<u>123</u>	Ahmed	01110025878
<u>124</u>	Ali	01225929785

- selects all Employees with a Name that does NOT start with "A"

```
SELECT *  
FROM Employee  
WHERE Ename NOTLIKE 'A%';
```

<u>Enum</u>	Ename	phone
<u>127</u>	Ola	0102457896

ORDER BY: Sorting the Results

33

- The column specified in the ORDER BY clause **does not** need to be included in the SELECT clause
- Null values are ordered as the lowest value

Employee

<u>Enum</u>	Ename	Salary	Gender
<u>123</u>	Ahmed	10000	M
<u>124</u>	Ali	5000	M
<u>127</u>	Ola	30000	F

```
SELECT Ename, Salary
FROM Employee
WHERE gender='M'
ORDER BY Salary DESC
```



Ename	Salary
Ahmed	10000
Ali	5000

```
SELECT Ename
FROM Employee
ORDER BY Salary DESC
```



Ename
Ola
Ahmed
Ali

ORDER BY Cont

34

□ Order by several columns

```
SELECT Lname, Fname, Salary  
FROM Employee  
WHERE Sex='F'  
ORDER BY Fname, Lname
```

```
SELECT PName, Price, Manufacturer  
FROM Product  
WHERE Category='gizmo' AND Price > 50  
ORDER BY Price ASC, Pname DESC
```

SELECT : with ALIAS

35

Employee

<u>Enum</u>	Ename	phone
<u>123</u>	Ahmed	01110025878
<u>124</u>	Ali	01225929785
<u>127</u>	Ola	0102457896

- Alias is an alternative column or table name

SELECT Ename as **Name**, Enum as **Employee ID**
FROM Employee

Name	Employee ID
Ahmed	123
Ali	124
Ola	127

SELECT Cust.Customer_Name as **Name**, Cust.Customer_address
FROM Customer **Cust**
WHERE Customer_Name= 'Home Furnishings';

Question

36

1. Create a relation “MyCustomers”
 - ▣ MyCustomers(customer_id, customer_name, city)
2. In your newly created “MyCustomers” table
 - ▣ Insert two customers with names “Ahmed” and “Mohamed” who live in “Paris”
 - ▣ Update all customers living in Paris, set their name to your name
 - ▣ Insert a new customer with your name and living in Cairo
 - ▣ Update the new customer with your father’s name
 - ▣ Update the new customer city to Alexandria