



VIT[®]

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

CSE2004

Database Management System

Slot: L15+L16

Week -IV

Venue: AB1-605B

SQL Built in & Aggregate Functions

Prof. Tulasi Prasad Sariki

SCSE, VIT, Chennai

www.learnersdesk.weebly.com

SELECT

- Querying data from a table is one of the most common tasks that you have to deal with on a regular basis.
- To query data from one or more tables, you use the SQL **SELECT** statement.
- SQL **SELECT** statement syntax

```
SELECT column1,column2...  
FROM table1,table2...  
WHERE filter_condition  
GROUP BY column1  
HAVING group_condition  
ORDER BY column1,column2 ASC | DESC
```



SELECT

- **SELECT:** specifies which columns in database tables to include in the result or result set.
- **FROM:** specifies the tables that you want to query data from. You can also use the FROM clause to join multiple tables.
- **WHERE:** filters unwanted records or rows in the result.
- **GROUP BY:** groups rows by common column values.
- **HAVING:** uses with the GROUP BY clause to filter unwanted groups.
- **ORDER BY:** sorts the result by one or more columns.



Select Examples

```
SELECT * FROM Employees;
```

```
SELECT Lastname FROM Employees
```

```
SELECT lastName, firstName FROM Employees
```

```
SELECT 1 + 1;
```

```
SELECT 2*3 from dual;
```

```
SELECT CONCAT(LastName,', ', FirstName) AS fullname FROM  
employees
```



Select (Distinct, All)

```
SELECT DISTINCT column_1, column2 FROM table;
```

```
SELECT city FROM employees;
```

```
SELECT DISTINCT city FROM employees;
```

```
SELECT DISTINCT city, country FROM employees;
```

```
SELECT COUNT(DISTINCT city) FROM employees;
```

```
SELECT SUM(DISTINCT unitprice) FROM Products;
```

```
SELECT ALL column_1, column_2 FROM table; (SELECT = SELECT ALL)
```



Select - Where

SELECT column_1, column_2 **FROM** table **WHERE** condition;

Comparison Operators : =, > , <, >= , <= and <>

- **SELECT** lastname, firstname, title **FROM** employees **WHERE** lastname = 'King';
- **SELECT** lastname, firstname, title **FROM** employees **WHERE** country <> 'USA';
- **SELECT** lastname, firstname, title, country, **DATE**(hiredate) **FROM** employees **WHERE** hiredate < '1993-01-01'.
- **SELECT** lastname, firstname, title, country, **DATE**(hiredate) **FROM** employees **WHERE** hiredate > '1993-01-01'.



Select - Where

SELECT column_1, column_2 **FROM** table **WHERE** condition;

Logical Operators : **AND**, **OR**, and **NOT**

- **SELECT** lastname, firstname, title, country, **DATE**(hiredate) **FROM** employees **WHERE** hiredate < '1993-01-01' **AND** country = 'USA'
- **SELECT** lastname, firstname, title, country, **DATE**(hiredate) **FROM** employees **WHERE** hiredate > '1993-01-01' **OR** country = 'USA'
- **SELECT** lastname, firstname, title, country, **DATE**(hiredate) **FROM** employees **WHERE NOT** (city = 'London' **OR** city = 'Seattle')

Note : Besides those operators, you can also use the **BETWEEN**, **IN**, **LIKE**, **EXISTS**, and **IS** operators in the **WHERE** clause.



SELECT - ORDER BY

SELECT column1, column2 **FROM** table **ORDER BY** (expr | column) **ASC**,
(expr | column) **DESC**;

- **SELECT** lastname, firstname **FROM** employees **ORDER BY** lastname;
- **SELECT** lastname, firstname **FROM** employees **ORDER BY** lastname **DESC**, firstname **ASC**;
- **SELECT** **CONCAT**(lastname, ',', firstname) fullname **FROM** employees **ORDER BY** **CONCAT**(lastname, ',', firstname);
- **SELECT** lastname, firstname, **DATE**(hiredate) **FROM** employees **ORDER BY** 3 **DESC**;



SELECT - Alias (Column/Table)

- **SELECT** productName **AS** product, unitPrice **AS** price **FROM** products **WHERE** unitPrice >50;
- **SELECT** productName product, unitPrice “unit Price” **FROM** products **WHERE** unitPrice >50;
- **SELECT** P.productName **FROM** products P **WHERE** P.unitPrice >50;



SELECT - BETWEEN

- **SELECT** column_1, column_2 **FROM** table **WHERE** (expr | column) **BETWEEN** lower_value **AND** upper_value;
- **SELECT** productName, unitPrice **FROM** products **WHERE** unitPrice **BETWEEN** 18 **AND** 19;
- **SELECT** productName, unitPrice **FROM** products **WHERE** unitPrice >=18 **AND** unitPrice <=19 ;
- **SELECT** lastname, firstname, birthdate **FROM** employees **WHERE** birthdate **BETWEEN** '1948-01-01' **AND** '1960-01-01';
- **SELECT** productName, unitPrice **FROM** products **WHERE** unitPrice **NOT BETWEEN** 18 **AND** 19;



SELECT - IN

- **SELECT** column_1, column_2 **FROM** table **WHERE** (expr | column) **IN** (value1 , value2, ...);
- **SELECT** productName, unitPrice **FROM** products **WHERE** unitPrice **IN** (18 , 19, 20)
- or
- **SELECT** productName, unitPrice **FROM** products **WHERE** unitPrice = 18 **OR** unitPrice = 19 **OR** unitPrice = 20;
- **SELECT** productName, unitPrice **FROM** products **WHERE** unitPrice **NOT IN** (18 , 19, 20)



SELECT - LIKE, IS

- **SELECT** column1, column2 **FROM** table **WHERE** column **LIKE** pattern;
- **SELECT** lastname, firstname **FROM** employees **WHERE** lastname **LIKE** 'D%'
- **SELECT** lastname, firstname **FROM** employees **WHERE** lastname **LIKE** '%t'
- **SELECT** lastname, firstname **FROM** employees **WHERE** lastname **LIKE** '%D%'
- **SELECT** lastname, firstname **FROM** employees **WHERE** lastname **LIKE** '_a%'
- **SELECT** lastname, firstname **FROM** employees **WHERE** lastname **LIKE** '_a_'
- **SELECT** companyName, fax **FROM** suppliers **WHERE** fax **IS** NULL;
- **SELECT** companyName, fax **FROM** suppliers **WHERE** fax **IS** NOT NULL;



SELECT - GROUP BY (MIN, MAX, AVG, COUNT)

- **SELECT** categoryid, **SUM**(unitsinstock) **FROM** products **GROUP BY** categoryid;
- **SELECT** categoryid, **COUNT**(productid) **FROM** products **GROUP BY** categoryid;
- **SELECT** categoryid, **FLOOR**(**AVG**(unitsinstock)) **FROM** products **GROUP BY** categoryid;
- **SELECT** categoryid, **MIN**(unitsinstock), **MAX**(unitsinstock) **FROM** products **GROUP BY** categoryid;
- **SELECT** categoryid, **COUNT**(productid) **FROM** products **GROUP BY** categoryid **ORDER BY** **COUNT**(productid) **DESC**;



SELECT - HAVING

- **SELECT column1, column2, aggregate_function(expr) FROM table
GROUP BY column1 HAVING condition;**
- **SELECTorderid, SUM(unitPrice * quantity) Total FROM orderdetails
GROUP BYorderid HAVING total > 12000;**
- **SELECTorderid, COUNT(productID) products FROM orderdetails
GROUP BYorderid HAVING products > 5;**





Thank You!