IT Department

Conditions and DML Statements

Soma Soleiman Zadeh

Database Systems I (IT 215)

Fall 2025 - 2026

Week 11

December 15, 2025

Outline

- Conditions in **SELECT** Query
  - **IF-ELSE**
  - **CASE**
- Remaining **DML** Statements
  - **INSERT** Statement
  - **DELETE** Statement
  - **UPDATE** Statement

Conditions in SELECT Query

- Rather than **WHERE** and **HAVING** clauses, SQL has **IF-ELSE** and **CASE** to add conditions.
- Syntax of Using **IF-ELSE** in **SELECT** Statement:

```
SELECT  IF (condition, true_statement, false_statement)
FROM  table_name;
```

IF-ELSE in SELECT Query

- By having the **Teacher** table, write a query to show output according to the given conditions:
  - **High Salary**: Salary > 3000
  - **Low Salary**: Salary <= 3000

Teacher

teacherID	teacherName	deptName	teacherRank	salary
BI01	Adams	Biology	Lecturer	3000
CS01	Byrne	IT	Assistant Prof	2000
CS02	Smith	IT	Assistant Lec	1400
CS03	John	IT	Lecturer	1800
EN01	Smith	English	Professor	5000
EN02	Leonardo	English	Assistant Lec	1500
EN03	Kate	English	Lecturer	1700
HI01	Kim	History	Assistant Prof	2500
MA01	Julia	Mathematics	Assistant Lec	1100
SP01	Maria	Sport	Professor	4000
SP02	Sarah	Sport	Lecturer	2000

IF-ELSE in SELECT Query

```
SELECT  teacherName, Salary, if (salary>3000, 'High Salary', 'Low Salary') AS Status
FROM  Teacher;
```

teacherName	salary	Status
Adams	3000	Low Salary
Byrne	2000	Low Salary
Smith	1400	Low Salary
John	1800	Low Salary
Smith	5000	High Salary
Leonardo	1500	Low Salary
Kate	1700	Low Salary
Kim	2500	Low Salary
Julia	1100	Low Salary
Maria	4000	High Salary
Sarah	2000	Low Salary

Conditions in SELECT Query

- For **nested IF-ELSE** situations, another **IF-ELSE** will be written in the place of the **false_statement**.

```
SELECT  IF (condition, true_statement,
            IF (condition, true_statement, false_statement) )
FROM  table_name;
```

Nested IF-ELSE in SELECT Query

- By having **Teacher** table, write a query to show output according to the given conditions:
  - **High Salary**: Salary > 3000
  - **Medium Salary**: Salary >= 2000 and < = 3000
  - **Low Salary**: Salary < 2000

Teacher

teacherID	teacherName	deptName	teacherRank	salary
BI01	Adams	Biology	Lecturer	3000
CS01	Byrne	IT	Assistant Prof	2000
CS02	Smith	IT	Assistant Lec	1400
CS03	John	IT	Lecturer	1800
EN01	Smith	English	Professor	5000
EN02	Leonardo	English	Assistant Lec	1500
EN03	Kate	English	Lecturer	1700
HI01	Kim	History	Assistant Prof	2500
MA01	Julia	Mathematics	Assistant Lec	1100
SP01	Maria	Sport	Professor	4000
SP02	Sarah	Sport	Lecturer	2000

Nested IF-ELSE in SELECT Query

```
SELECT  teacherName, Salary, if (salary>3000, 'High Salary',
                    if (salary BETWEEN 2000 AND 3000, 'Medium Salary',  'Low Salary') )
   AS Status
FROM  Teacher;
```

teacherName	salary	Status
Adams	3000	Medium Salary
Byrne	2000	Medium Salary
Smith	1400	Low Salary
John	1800	Low Salary
Smith	5000	High Salary
Leonardo	1500	Low Salary
Kate	1700	Low Salary
Kim	2500	Medium Salary
Julia	1100	Low Salary
Maria	4000	High Salary
Sarah	2000	Medium Salary

Conditions in SELECT Query - CASE

- In SQL, **CASE** statement acts the same as **IF-ELSE** statement.
- It can specify ranges as conditions.
- Benefits of using **CASE** statement:
  - Simplifying complex queries
  - Enhancing Readability of SQL Queries

DML Statements

INSERT, DELETE, UPDATE

SQL DML – Manipulating the Database

- The SQL **DML** statements are:

SELECT	To query (request) data in the database
UPDATE	To update data in a table
INSERT	To insert data in a table
DELETE	To delete data from a table

Modifying Data Inside the Table (DML Statements

- There are three main statements for changing the data of tables:

1. **INSERT**
   - Inserting a record or records into a table by entering values manually
   - Getting data from a table (by using SELECT) and inserting it into another table.
2. **DELETE**
   - Deleting a record or records from a table.
3. **UPDATE**
   - Changing some field values of records
   - **Example –** Updating a student's phone number to a new number
     Increasing all faculties' salaries by $500.

INSERT Statement

- The **INSERT** operator is used to enter new records into a table.

INSERT INTO *table_name* (column names) **VALUES** (value. . .);

Note: The column names are optional if we are inserting values for all columns in their proper order.

Inserting One Record into a Table

- Add '**Accounting**' department, with deptCode equal to "**AC09**", and office room number '**A110'** and budget $3000 in department table.

department (deptName, deptCode, office, budget)

insert into department values ('Accounting', 'AC09' , 'A110', 3000)

OR

insert into department (deptName, deptCode, office, budget)
values ('Accounting', 'AC09' , 'A110', 3000)

Which INSERT Statement is NOT Correct?

```
insert into Student (stuId, firstName) values ('S1025', 'Bob');
insert into Student (lastName, major) values ('James', 'History');
```

Student

stuID	firstName	lastName	major	credits
S1001	Tom	Smith	History	90
S1002	Ann	Chin	Mathematics	36
S1004	Smith	Jack	English	75
S1005	Lee	Perry	History	3
S1007	Streep	Sarah	English	81
S1010	Burns	Edward	Biology	63
S1011	Roberts	Mike	English	66
S1012	Damon	Tom	IT	90
S1013	McCarthy	Owen	Mathematics	27
S1015	Jones	Mary	Sport	42
S1017	Ford	Jennifer	History	45
S1018	Nolan	Ryan	English	50
S1020	Rivera	Jane	IT	15

DELETE Statement

- The **DELETE** statement is used to erase records.
- The number of records deleted may be zero, one, or many, depending on how many satisfy the condition.

```
DELETE FROM table_name
WHERE conditions;
```

DELETE Statement Example

- Delete all records in **Student** table.

```
DELETE FROM Student;
```

- Delete all information about students in **'English'** department.

```
DELETE FROM Student
WHERE major = 'English';
```

Student

stuID	firstName	lastName	major	credits
S1001	Tom	Smith	History	90
S1002	Ann	Chin	Mathematics	36
S1004	Smith	Jack	English	75
S1005	Lee	Perry	History	3
S1007	Streep	Sarah	English	81
S1010	Burns	Edward	Biology	63
S1011	Roberts	Mike	English	66
S1012	Damon	Tom	IT	90
S1013	McCarthy	Owen	Mathematics	27
S1015	Jones	Mary	Sport	42
S1017	Ford	Jennifer	History	45
S1018	Nolan	Ryan	English	50
S1020	Rivera	Jane	IT	15

UPDATE Statement

- The **UPDATE** operator is used to change values in the current records in a table.
- **UPDATE** is used on one table at a time, and can change zero, one, or many records.

```
UPDATE      table-name
SET         column_name = expression
            [,column_name = expression] . . .
[WHERE      condition];
```

UPDATE Statement

UPDATE Teacher
SET teacherRank = 'Lecturer'
WHERE teacherRank = ' Assistant Lec';

Teacher

teacherID	teacherName	deptName	teacherRank	salary
BI01	Adams	Biology	Lecturer	3000
CS01	Byrne	IT	Assistant Prof	2000
CS02	Smith	IT	Lecturer	1400
CS03	John	IT	Lecturer	1800
EN01	Smith	English	Professor	5000
EN02	Leonardo	English	Lecturer	1500
EN03	Kate	English	Lecturer	1700
HI01	Kim	History	Assistant Prof	2500
MA01	Julia	Mathematics	Lecturer	1100
SP01	Maria	Sport	Professor	4000
SP02	Sarah	Sport	Lecturer	2000

UPDATE Example

- Change the department of a teacher named 'John' to **MIS** and his rank to **"Assistant Prof"**.

UPDATE Teacher
SET deptName = 'MIS' ,
teacherRank = 'Assistant Prof'
WHERE teacherName = ' John';

Teacher

teacherID	teacherName	deptName	teacherRank	salary
BI01	Adams	Biology	Lecturer	3000
CS01	Byrne	IT	Assistant Prof	2000
CS02	Smith	IT	Assistant Lec	1400
CS03	John	IT	Lecturer	1800
EN01	Smith	English	Professor	5000
EN02	Leonardo	English	Assistant Lec	1500
EN03	Kate	English	Lecturer	1700
HI01	Kim	History	Assistant Prof	2500
MA01	Julia	Mathematics	Assistant Lec	1100
SP01	Maria	Sport	Professor	4000
SP02	Sarah	Sport	Lecturer	2000

UPDATE Example

- Add 10 to students' credits in the "IT" or "Biology" departments.

Student

stuID	firstName	lastName	major	credits
S1001	Tom	Smith	History	90
S1002	Ann	Chin	Mathematics	36
S1004	Smith	Jack	English	75
S1005	Lee	Perry	History	3
S1007	Streep	Sarah	English	81
S1010	Burns	Edward	Biology	63
S1011	Roberts	Mike	English	66
S1012	Damon	Tom	IT	90
S1013	McCarthy	Owen	Mathematics	27
S1015	Jones	Mary	Sport	42
S1017	Ford	Jennifer	History	45
S1018	Nolan	Ryan	English	50
S1020	Rivera	Jane	IT	15