

Database Management Systems

SQL (4)

Topics

- ▶ Altering Table Schema
- ▶ Queries with
 - ▶ BETWEEN, NOT BETWEEN
 - ▶ IN, NOT IN
- ▶ Subqueries with
 - ▶ Select, update, insert and delete
 - ▶ Aggregate functions
 - ▶ IN
 - ▶ EXISTS, NOT EXISTS

Altering Schema of a Table

- ▶ **ALTER TABLE** is used to
 - ▶ Rename a table
 - ▶ Add a column
 - ▶ Remove a column
 - ▶ Change data type of a column
 - ▶ Change restrictions

ALTER TABLE

- ▶ Change the name of a table using ALTER TABLE
- ▶ ALTER TABLE cur_table_name RENAME new_table_name
- ▶ E.g. ALTER TABLE student RENAME Grad_Students

ALTER TABLE

- ▶ Change the data type of an attribute using ALTER TABLE
- ▶ For example a numeric attribute can be changed to a varchar attribute
- ▶ If you change a character attribute to a numeric attribute you may receive error if the column contains data (data should be convertible)

ALTER TABLE

ALTER TABLE `table_name` **MODIFY** `column_name` `data_type`

e.g.

CREATE TABLE student

(

student_ID Integer primary key,

name varchar,

address varchar

)

ALTER TABLE student **MODIFY** name char(30)

ALTER TABLE

- ▶ Change the name of an attribute
- ▶ **ALTER TABLE** table_name **CHANGE** old_column_name
new_column_name

e.g. **ALTER TABLE** student **CHANGE** name student_name

ALTER TABLE

- ▶ Delete a column using ALTER TABLE
- ▶ **ALTER TABLE** table_name **DROP COLUMN** column_name

e.g. **ALTER TABLE** student **DROP COLUMN** telephone

ALTER TABLE

- ▶ Add a new column to a table
- ▶ **ALTER TABLE** table_name **ADD** column_name data_type restrictions
- ▶ **ALTER TABLE** student **ADD** date_of_birth date **NOT NULL**

BETWEEN , NOT BETWEEN

- ▶ The BETWEEN operator is used to select values within a range.
- ▶ `SELECT column_name(s)`
`FROM table_name`
`WHERE column_name BETWEEN value1 AND value2`
- ▶ `SELECT * FROM Products`
`WHERE Price BETWEEN 10 AND 20`
- ▶ `SELECT * FROM Products`
`WHERE Price NOT BETWEEN 10 AND 20`

IN, NOT IN

- ▶ IN clause is used to match a column value with a value from a list.
- ▶ IN helps in replacing many OR statements

- ▶ `SELECT * FROM employee
WHERE salary IN (250, 220, 170)`
- ▶ `SELECT * FROM employee
WHERE salary NOT IN (250, 220, 170)`

Subqueries

- ▶ A Subquery or Inner query or Nested query is a query within another SQL query and is written inside the WHERE clause.
- ▶ A subquery is used to return data that will be used in the main query as a condition.

Using a Subquery to Solve a Problem

- ▶ Who has a salary greater than Smith's?
- ▶ **Main query:** Which employees have salaries greater than smith's salary
- ▶ **Subquery:** What is Smith's salary?

Using Subquery

```
SELECT last_name
FROM employees
WHERE salary >
      (SELECT salary
       FROM employees
       WHERE last_name = 'Smith');
```

Using Subquery in INSERT and UPDATE

Copy all records from *temp* table to *Product* table

```
INSERT INTO Product  
SELECT * FROM temp
```

Using Subquery in INSERT and UPDATE

- ▶ Inserts the records of students studying in CENG, IE, ECE, MECE or EE departments into Engineering table

```
INSERT INTO Engineering  
(SELECT * FROM student  
WHERE Dept IN ('CENG', 'IE', 'ECE', 'MECE', 'EE' ))
```


Using Subquery in INSERT and UPDATE

Set price of the product with product ID = 100 to be the average price of all orders of that product

```
UPDATE Product
SET price = (SELECT avg(price)
             FROM Order
             WHERE Product_id = 100)
WHERE Product_id = 100
```

DELETE Subquery Examples

Delete all products with an order price of less than 1000

```
DELETE * FROM product  
WHERE Product_id IN  
(SELECT Pid  
FROM Orders  
WHERE price < 1000 )
```

Using Aggregate Functions in a Subquery

Find the name and job-id and salary of the employee who has the minimum salary

```
SELECT last_name, job_id, salary
FROM employees
WHERE salary =
        (SELECT MIN(salary)
         FROM employees);
```

EXISTS, NOT EXISTS

EXISTS is used to verify if any record with the given requirements exists in a table. For example, is there any student with a grade of 90 or higher from CENG356?

```
SELECT student_id, student_name
FROM student
WHERE EXISTS (SELECT *
              FROM grades
              WHERE student.student_id = grades.student_id AND
                 grades.course_code = 'CENG356' AND
                 grades.score >= 90 );
```

EXISTS, NOT EXISTS

Find names of students who have not taken CENG356 so far

```
SELECT student_id, student_name
```

```
FROM student
```

```
WHERE NOT EXISTS (SELECT *
```

```
    FROM grades
```

```
    WHERE student.student_id = grades.student_id AND
```

```
        grades.course_code = 'CENG356')
```

EXISTS, NOT EXISTS

Delete all customers who have put no order

```
DELETE FROM customers
WHERE NOT EXISTS (SELECT *
                  FROM orders
                  WHERE customers.Customer_id = orders.Customer_id)
```

Questions

