##### **ASSIGNMENT-3 (SQL | Subqueries and DDL-DML Commands)**

**Instructions:**

* There are multiple questions.
* Use **chinook**database to answer the questions.
* Please paste your answers (statements) as a plain text using the related topic and question number.  
  Good Luck!

**Questions:**

**Single-Row Subqueries:**

1. Write a query to find the maximum duration among the tracks. Your query should include TrackId, Name, Milliseconds.

ÇÖZÜM-1

SELECT TrackId, Name, MAX(Milliseconds) AS Max\_Duration

FROM tracks;

ÇÖZÜM-2

SELECT TrackId, Name, (SELECT MAX(Milliseconds) FROM tracks) AS Max\_Duration

FROM tracks

ORDER BY Milliseconds DESC

LIMIT 1;

1. Write a query to find the minimum duration among the tracks. Your query should include TrackId, Name, Milliseconds.

ÇÖZÜM-1

SELECT TrackId, Name, MIN(Milliseconds) AS Min\_Duration

FROM tracks;

ÇÖZÜM-2

SELECT TrackId, Name, (SELECT MIN(Milliseconds) FROM tracks) AS MIN\_Duration

FROM tracks

ORDER BY Milliseconds ASC

LIMIT 1;

1. Write a query to find the tracks whose bytes are higher than the average of the bytes of all tracks. Your query should include TrackId, Name, Bytes, general average and should be ordered by Bytes from highest to lowest. General average is the average of all tracks. General average should repeat in every row. (Hint: You need to use two subqueries)

SELECT TrackId, Name, Bytes, (SELECT AVG(Bytes) FROM tracks) AS General\_Average

FROM tracks

WHERE Bytes > (SELECT AVG(Bytes) FROM tracks)

ORDER BY Bytes DESC;

**Multiple-Row Subqueries:**

1. Write a query that returns the customers whose sales representatives are Jane Peacock and Margaret Park. Your query should include CustomerID, First Name and Last Name of the customers.

SELECT CustomerId, FirstName, LastName

FROM customers

WHERE SupportRepId

IN (SELECT EmployeeId

FROM Employees

WHERE Firstname = 'Jane' AND LastName = 'Peacock'

OR Firstname = 'Margaret' AND LastName = 'Park');

1. Rewrite the subquery above using the JOIN.

SELECT

customers.CustomerId,

customers.FirstName,

customers.LastName

FROM customers

INNER JOIN employees

ON customers.SupportRepId = employees.EmployeeId

WHERE EmployeeId = 3 OR EmployeeId = 4;

**DDL (CREATE, ALTER, DELETE) and DML (SELECT, INSERT, UPDATE, DELETE) Statements**

1. Create a table inside the chinook database. This table tracks the employees’ courses inside the organization. Your table should have the followings features:

               Name of the table: courses  
               Columns:  
                    CourseId (Primary Key)  
                    CourseName (Cannot be null)  
                    EmployeeId (Foreign Key - Refers to EmployeeId of employees table )   
                    CoursePrice

DROP TABLE IF EXISTS courses;

CREATE TABLE courses

(CourseId INTEGER PRIMARY KEY,

CourseName VARCHAR(30) NOT NULL,

EmployeeId INTEGER,

CoursePrice INTEGER,

FOREIGN KEY (EmployeeId)

REFERENCES employees(EmployeeId));  
  
      2. Insert at least 10 rows into the *courses* table. Your EmployeeId should contain the values of the *EmployeeId*column on the employees table. You’re free to choose any values for other columns (CourseId, CourseName, CoursePrice)

INSERT INTO courses(CourseID, CourseName, CoursePrice, EmployeeId)

VALUES (1, 'Yönetişim', 100, 1),

(2, 'Muhasebe', 200, 2),

(3, 'Görüşme\_Teknikleri', 130, 3),

(4, 'İkna\_Yöntemleri', 140, 4),

(5, 'Jest\_Mimikler', 150, 5),

(6, 'GIT-GITHUB', 160, 6),

(7, 'Computer\_Fundamentals', 125, 7),

(8, 'Computational\_Thinking', 135, 8),

(9, 'Scratch', 145, 7),

(10, 'Python', 155, 8);

1. Delete the last row of your *courses* table.

DELETE FROM courses

WHERE CourseId = (SELECT MAX(CourseId) FROM courses)

1. Add a new column to your *courses* table named *StartDate*. The new column cannot be null.

ALTER TABLE courses

ADD StartDate DATETIME;

1. Delete the *CoursePrice* column.

ALTER TABLE courses

DROP COLUMN CoursePrice;

1. Delete the *courses* table.

DROP TABLE courses;

DROP TABLE IF EXISTS vacation\_plan;  
CREATE TABLE vacation\_plan  
(place\_id INT PRIMARY KEY,  
Country VARCHAR(20),  
hotel\_name CHAR (20) NOT NULL,  
EmployeeId INT,  
Vacation\_lenght INT,  
budget REAL,  
FOREIGN KEY (EmployeeId) REFERENCES employees(EmployeeId));  
INSERT INTO vacation\_plan  
(place\_id, country, hotel\_name, EmployeeId, vacation\_lenght, budget)  
VALUES  
(1, ‘Canada’, ‘Hilton’, 1, 5, 10000),  
(2, ‘USA’, ‘Sheraton’, 1, 5, 10000),  
(3, ‘Turkey’, ‘Erzincan\_plus’, 1, 8, 500);ALTER TABLE vacation\_plan  
ADD city TEXT;ALTER TABLE vacation\_plan  
RENAME TO new\_vacation\_plan;