|  |  |
| --- | --- |
| **Topic** | Oracle SQL Language Fundamentals I |
| **Document Name** | SQL03-EX-01-05 |
|  |  |

## Exercise SQL03-EX-01:

**Definiton :** Write followig SQL queries:

* Add a colum to employees table named MAX\_SALARY.
* Update MAX\_SALARY with maximum salary amount with subquery.
* Delete employee who have minimum salary using subquery.

**SQL:**

ALTER TABLE EMPLOYEES

ADD MAX\_SALARY NUMBER;

UPDATE EMPLOYEES

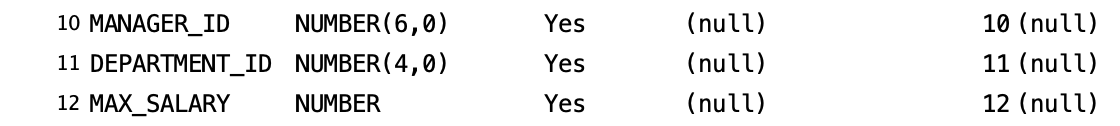
SET MAX\_SALARY = (SELECT MAX(salary) FROM employees);

DELETE FROM EMPLOYEES

WHERE SALARY = (SELECT MIN(SALARY) FROM EMPLOYEES);

**Screenshot:**

* MAX\_SALARY column eklendi ve tüm column max salary olan 24000 ile update edildi.

****

* EMPLOYEE\_ID 132 olanın salary en düşüktü o satır silindi

**metin, yazı tipi, ekran görüntüsü, makbuz içeren bir resim

Açıklama otomatik olarak oluşturuldu**

## Exercise SQL03-EX-02:

**Definiton :** Write followig SQL queries:

* Define index (named DPR\_NAME\_IDX) on DEPARTMENT\_NAME column of DEPARTMENTS table.
* Define constraint (named CNSTR\_SALARY) on employee salary. (Salary must be between 1000$ and 100.000$)
* Drop defined index.
* Enable, disable, drop defined constraint.

**SQL:**

CREATE INDEX DPR\_NAME\_IDX

ON DEPARTMENTS (DEPARTMENT\_NAME);

ALTER TABLE EMPLOYEES

ADD CONSTRAINT CNSTR\_SALARY

CHECK (SALARY BETWEEN 1000 AND 100000);

DROP INDEX DPR\_NAME\_IDX;

ALTER TABLE EMPLOYEES

DISABLE CONSTRAINT CNSTR\_SALARY;

ALTER TABLE EMPLOYEES

ENABLE CONSTRAINT CNSTR\_SALARY;

ALTER TABLE EMPLOYEES

DROP CONSTRAINT CNSTR\_SALARY;

**Screenshot:**

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

## Exercise SQL03-EX-03:

**Definiton :** Create a table from EMPLOYEES with distinct department\_id column. Add department\_name to that table. With DEPARTMENTS table, update department\_name for included department\_ids and insert department\_id and department\_name values for not included rows. Use MERGE keyword.

**SQL:**

CREATE TABLE MY\_DISNT\_TABLE AS SELECT DEPARTMENT\_ID, DEPARTMENT\_NAME FROM DEPARTMENTS;

MERGE INTO MY\_DISNT\_TABLE m

USING DEPARTMENTS d ON (m.DEPARTMENT\_ID = d.DEPARTMENT\_ID)

WHEN MATCHED THEN UPDATE SET m.DEPARTMENT\_NAME = d.DEPARTMENT\_NAME

WHEN NOT MATCHED THEN INSERT (DEPARTMENT\_ID, DEPARTMENT\_NAME) VALUES (d.DEPARTMENT\_ID,

d.DEPARTMENT\_NAME);

**Screenshot:**

metin, ekran görüntüsü, yazı tipi, beyaz içeren bir resim

Açıklama otomatik olarak oluşturuldu

## Exercise SQL03-EX-04:

**Definiton :** Using **WITH** keyword, do following jobs:

* Firstly select first\_name, last\_name, job\_id, department\_id from employees table whoes job\_id starts with ‘S’.
* Additionally select job\_title and min-max salary amount.
* Add department\_name to that query.
* Lastly concat first\_name and last\_name with space as full\_name alias and list with other selected columns.

**SQL:**

SELECT

CONCAT(e.first\_name, ' ') || e.last\_name AS full\_name,

e.first\_name,

e.last\_name,

e.job\_id,

e.department\_id,

j.job\_title,

j.min\_salary,

j.max\_salary,

d.department\_name

FROM

employees e

JOIN

jobs j ON e.job\_id = j.job\_id

JOIN

departments d ON e.department\_id = d.department\_id

WHERE

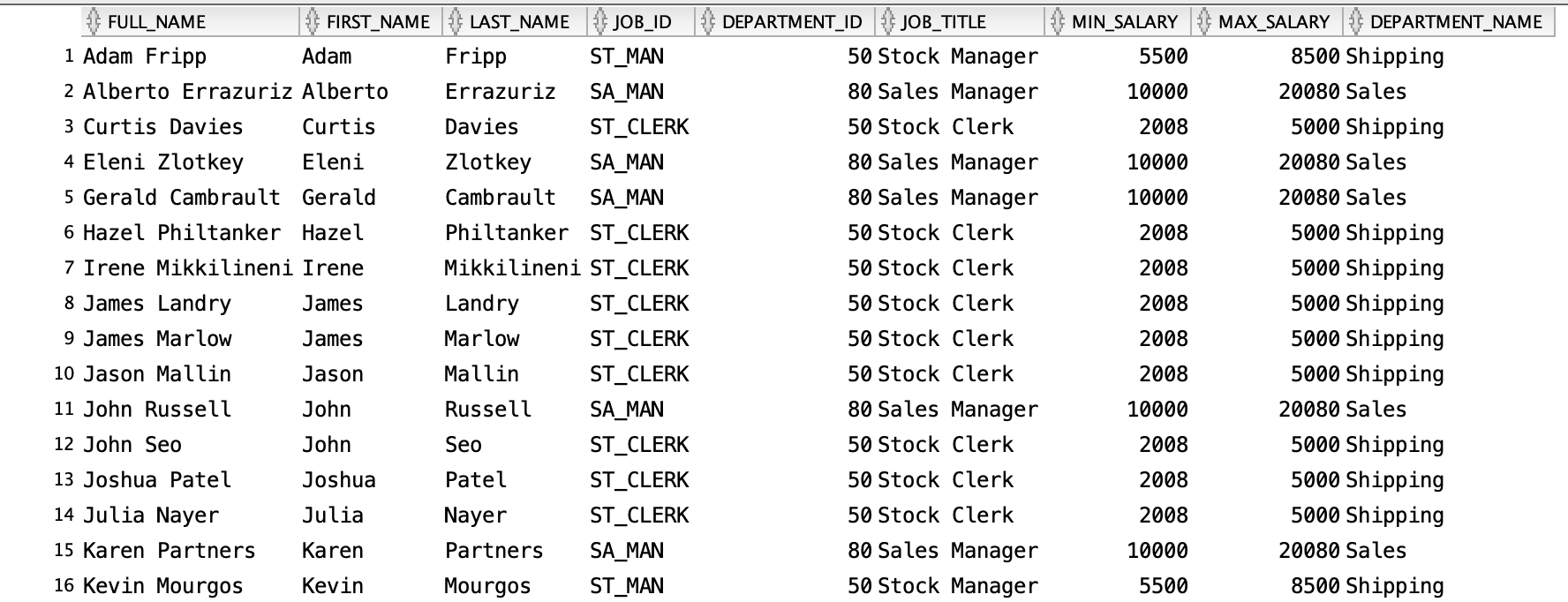
e.job\_id LIKE 'S%'

ORDER BY

full\_name;

**Screenshot:**

WITH ile yapmaya çalıştığımda sürekli SQL sorgusunu böyle bitiremezsin diye bir hata aldım. Bundan dolayı bu query ile yaptım



## Exercise SQL03-EX-05:

**Definiton :** Search for COMMIT and ROLLBACK keywords and explain them.

**COMMIT** and **ROLLBACK** are two essential SQL keywords used to manage database transactions.

**COMMIT:** The COMMIT statement permanently saves the changes made by the current transaction to the database. Once a COMMIT is issued, all changes within the transaction become permanent. The transaction ends successfully.

BEGIN TRANSACTION;

UPDATE customers SET credit\_limit = 10000 WHERE customer\_id = 123;

INSERT INTOorders (customer\_id, order\_date) VALUES (123, GETDATE());

COMMIT;

**The COMMIT comand statement makes these changes permanent**

**ROLLBACK**: The ROLLBACK statement undoes all changes made by the current transaction. The database is restored to its state before the transaction began. The transaction is terminated unsuccessfully.