|  |  |
| --- | --- |
| **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\_VALUE NUMBER;

UPDATE EMPLOYEES

SET

MAX\_SALARY = (SELECT MAX(SALARY) FROM EMPLOYEES) ;

DELETE FROM EMPLOYEES WHERE EMPLOYEE\_ID = (SELECT EMPLOYEE\_ID FROM EMPLOYEES WHERE SALARY = (SELECT MIN(SALARY) FROM EMPLOYEES));

**Screenshot:**

metin, ekran görüntüsü, sayı, numara, çizgi 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 >= 1000 AND salary <= 100000);

DROP INDEX DPR\_NAME\_IDX;

ALTER TABLE employees

ENABLE CONSTRAINT CNSTR\_SALARY;

ALTER TABLE employees

DISABLE CONSTRAINT CNSTR\_SALARY;

ALTER TABLE employees

DROP CONSTRAINT CNSTR\_SALARY;

## 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 DISTINCT\_DEPARTMENTS AS

SELECT DISTINCT department\_id

FROM EMPLOYEES;

ALTER TABLE DISTINCT\_DEPARTMENTS

ADD department\_name VARCHAR(100); -- Adjust the datatype and size as per your actual DEPARTMENTS table.

MERGE INTO DISTINCT\_DEPARTMENTS x

USING (SELECT department\_id, department\_name FROM departments) y

ON (x.department\_id = y.department\_id)

WHEN MATCHED THEN

UPDATE SET x.department\_name = y.department\_name

WHEN NOT MATCHED THEN

INSERT(x.department\_id, x.department\_name)

VALUES(y.department\_id, y.department\_name);

SELECT \* FROM DISTINCT\_DEPARTMENTS;

**Screenshot:**

metin, ekran görüntüsü, sayı, numara, yazı tipi 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:**

**Screenshot:**

## Exercise SQL03-EX-05:

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

**SQL:**

**COMMIT**: Saves the changes made in a transaction, making them permanent and visible to other users.

**ROLLBACK**: Undoes the changes made in a transaction, restoring the database to its state before the transaction started.

**Screenshot:**