|  |  |
| --- | --- |
| **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:**

UPDATE employees

SET max\_salary = (SELECT MAX(salary) FROM employees);

DELETE FROM employees

WHERE salary = (SELECT MIN(salary) FROM employees);

**Screenshot:**

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

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazı tipi, 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 BETWEEN 1000 AND 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 emp\_dept AS

SELECT DISTINCT department\_id

FROM employees;

ALTER TABLE emp\_dept

ADD (department\_name VARCHAR2(50));

MERGE INTO emp\_dept ed

USING departments d

ON (ed.department\_id = d.department\_id)

WHEN MATCHED THEN

UPDATE SET ed.department\_name = d.department\_name

WHEN NOT MATCHED THEN

INSERT (department\_id, department\_name)

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

**Screenshot:**

metin, yazı tipi, ekran görüntüsü, çizgi 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:**

WITH filtered\_employees AS (

SELECT employee\_id, first\_name, last\_name, job\_id, department\_id

FROM employees

WHERE job\_id LIKE 'S%'

),

job\_info AS (

SELECT job\_id, job\_title, MIN(salary) AS min\_salary, MAX(salary) AS max\_salary

FROM employees

GROUP BY job\_id, job\_title

),

department\_info AS (

SELECT department\_id, department\_name

FROM departments

)

SELECT

f.employee\_id,

f.first\_name,

f.last\_name,

f.job\_id,

f.department\_id,

j.job\_title,

j.min\_salary,

j.max\_salary,

d.department\_name,

f.first\_name || ' ' || f.last\_name AS full\_name

FROM

filtered\_employees f

LEFT JOIN job\_info j ON f.job\_id = j.job\_id

LEFT JOIN department\_info d ON f.department\_id = d.department\_id;

## Exercise SQL03-EX-05:

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

**COMMIT:** Saves all changes made during the current transaction permanently.

**ROLLBACK:** Undoes all changes made during the current transaction, reverting to the state before the transaction began.