|  |  |
| --- | --- |
| **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 HR.EMPLOYEES ADD MAX\_SALARY INTEGER;

UPDATE EMPLOYEES

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

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

**Screenshot:**

metin, makbuz, yazı tipi, ekran görüntüsü 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 CNSTR\_SALARY ON EMPLOYEES;

ALTER TABLE EMPLOYEES ENABLE CONSTRAINT CNSTR\_SALARY;

ALTER TABLE EMPLOYEES DISABLE CONSTRAINT CNSTR\_SALARY;

ALTER TABLE EMPLOYEES DROP CONSTRAINT CNSTR\_SALARY;

**Screenshot:**

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

MERGE INTO DISTINCT\_DEPARTMENTS DD

USING (

SELECT DISTINCT department\_id FROM EMPLOYEES WHERE department\_id IS NOT NULL

) E

ON (DD.department\_id = E.department\_id)

WHEN MATCHED THEN

UPDATE SET DD.department\_name = (

SELECT department\_name FROM DEPARTMENTS D WHERE D.department\_id = DD.department\_id

)

WHEN NOT MATCHED THEN

INSERT (department\_id, department\_name)

VALUES (E.department\_id, (

SELECT department\_name FROM DEPARTMENTS D WHERE D.department\_id = E.department\_id));

**Screenshot:**

metin, ekran görüntüsü, yazı tipi, sayı, numara 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 EmployeesJobS AS (

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

FROM employees

WHERE job\_id LIKE 'S%'

),

JobDetails AS (

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

FROM jobs

GROUP BY job\_id, job\_title

),

EmployeesWithDetails AS (

SELECT EJS.first\_name, EJS.last\_name, EJS.job\_id, EJS.department\_id, JD.job\_title, JD.min\_salary, JD.max\_salary

FROM EmployeesJobS EJS

JOIN JobDetails JD ON EJS.job\_id = JD.job\_id

)

SELECT

EWD.first\_name || ' ' || EWD.last\_name AS full\_name,

EWD.job\_id,

EWD.department\_id,

EWD.job\_title,

EWD.min\_salary,

EWD.max\_salary,

D.department\_name

FROM EmployeesWithDetails EWD

JOIN departments D ON EWD.department\_id = D.department\_id;

**Screenshot:**

metin, sayı, numara, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

## Exercise SQL03-EX-05:

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

COMMIT and ROLLBACK are performed on transactions. A transaction is the smallest unit of work that is performed against a database. Its a sequence of instructions in a logical order. A transaction can be performed manually by a programmer or it can be triggered using an automated program.

COMMIT is the SQL command that is used for storing changes performed by a transaction. When a COMMIT command is issued it saves all the changes since last COMMIT or ROLLBACK.

ROLLBACK is the SQL command that is used for reverting changes performed by a transaction. When a ROLLBACK command is issued it reverts all the changes since last COMMIT or ROLLBACK.