|  |  |
| --- | --- |
| **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

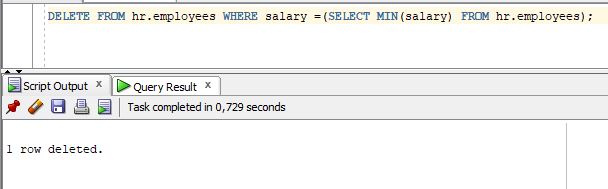
ADD (MAX\_SALARY INT);

UPDATE hr.employees

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

DELETE FROM hr.employees WHERE salary =(SELECT MIN(salary) FROM hr.employees);

**Screenshot:**



## 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 hr.departments(department\_name);

ALTER TABLE hr.employees ADD CONSTRAINT CNSTR\_SALARY

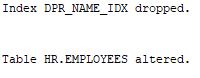
CHECK (SALARY BETWEEN 1000 AND 1000000);

ALTER TABLE HR.employees ENABLE CONSTRAINT CNSTR\_SALARY;

ALTER TABLE HR.employees DISABLE CONSTRAINT CNSTR\_SALARY;

ALTER TABLE HR.employees DROP CONSTRAINT CNSTR\_SALARY;

**Screenshot:**



## 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\_TABLE AS (SELECT department\_id FROM hr.employees);

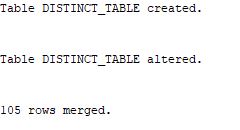
ALTER TABLE DISTINCT\_TABLE ADD (department\_name VARCHAR2(35));

MERGE INTO DISTINCT\_TABLE i USING (SELECT \* FROM HR.departments) d

ON (i.department\_id = d.department\_id) WHEN MATCHED THEN

UPDATE SET i.department\_name = d.department\_name;

**Screenshot:**



## 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 EMP\_JOBS AS (SELECT FIRST\_NAME,LAST\_NAME,JOB\_ID,DEPARTMENT\_ID FROM EMPLOYEES WHERE FIRST\_NAME LIKE 'S%'),

JOB\_MAX\_MIN AS ( SELECT JOB\_ID,JOB\_TITLE,MIN\_SALARY,MAX\_SALARY FROM JOBS),

DEP\_DET AS ( SELECT DEPARTMENT\_ID,DEPARTMENT\_NAME FROM DEPARTMENTS)

SELECT E.FIRST\_NAME,E.LAST\_NAME,E.JOB\_ID,

JM.JOB\_TITLE,JM.MAX\_SALARY,JM.MIN\_SALARY,

D.DEPARTMENT\_NAME,

CONCAT(E.FIRST\_NAME,E.LAST\_NAME) AS FULL\_NAME FROM EMP\_JOBS E

JOIN JOB\_MAX\_MIN JM ON E.JOB\_ID = JM.JOB\_ID

JOIN DEP\_DET D ON E.DEPARTMENT\_ID = D.DEPARTMENT\_ID;

**Screenshot :**

## Exercise SQL03-EX-05:

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

**SQL:**

"COMMIT," bir işlem veya işlemler kümesini kalıcı olarak veritabanına uygulama ve verilerin değişikliklerini veritabanında kaydetme işlemidir. Bir COMMIT işlemi, veritabanı yönetim sistemine yapılan tüm değişikliklerin tamamlanmış olduğunu ve geri alınamayacağını belirtir.

"ROLLBACK," yapılan bir işlem veya işlemleri geri almak ve veritabanında yapılan değişiklikleri geri döndürmek için kullanılan bir komuttur. ROLLBACK komutu, bir işlemi tamamlamadan veya bir hata durumunda geri almak ve veritabanını başlangıç durumuna geri getirmek için kullanılır.